



PHYSICS

ADVANCING FAAS APPLICATIONS IN THE CLOUD CONTINUUM

Presentation for Red Hat Research Days -- Nov 16, 2022

Main Speaker: Georgios Kousiouris (HUA)

Conversation Leader: Luis Tomás Bolívar (RHT) and Yiannis Georgiou (RYAX)



This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047



AGENDA

- ❑ Introduction to EU Projects
- ❑ PHYSICS EU Project
 - Consortium, WPs
 - Objectives
 - Components
 - Research topics and results
- ❑ Red Hat role and main technologies
- ❑ K8s Image Layer Scheduler: benefits of open source





PHYSICS

INTRODUCTION TO EU PROJECTS



This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047



3 main types

- ❖ Research and Innovation Actions (RIA)
 - ❖ Typically aiming to deliver prototypes and integrated environments (TRL 3-4-5)
- ❖ Innovation Actions (IA)
 - ❖ Start from existing prototypes (TRL3-4) and increase the TRL or target initial commercial services
- ❖ CSA actions
 - ❖ Support actions aiming at standardization, organization of dissemination etc.
- ❖ Budget Sizes: depends on the EC call for proposals



EC calls

- ❖ Centered around 7-year Framework Programs
 - ❖ FP7 (2007-2013), H2020 (2014-2020), Horizon Europe (2021-2027)
- ❖ Work programmes of 1-2 years ahead issued by the EC
- ❖ Consortia of partners organize and target a specific call:
 - ❖ PHYSICS RIA call example:
<https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/ict-40-2020>
 - ❖ Call has a specific high level scope: Smart cloud/edge continuum
 - ❖ Balance between academic and industrial partners (both large players and SMEs)
 - ❖ Balance between activities: R&D (if a RIA), piloting applications, innovation, dissemination, exploitation
- ❖ Work is split into Work Packages (WPs) and tasks inside WPs





PHYSICS

THE PHYSICS PROJECT

OPTIMIZED HYBRID SPACE-TIME SERVICE CONTINUUM IN FAAS



This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047



Challenges targeted by PHYSICS



Abstract usage of **service offerings** and **clusters across the Continuum**



Adaptation of code to new **serverless** computing paradigms



Investigation of **space** (location of execution)-**time** (duration of execution) in the continuum



Optimization of resource **selection** and operation (**global** and **local** level)



Multiple Exploitation Channels and **Reusable** Artefacts

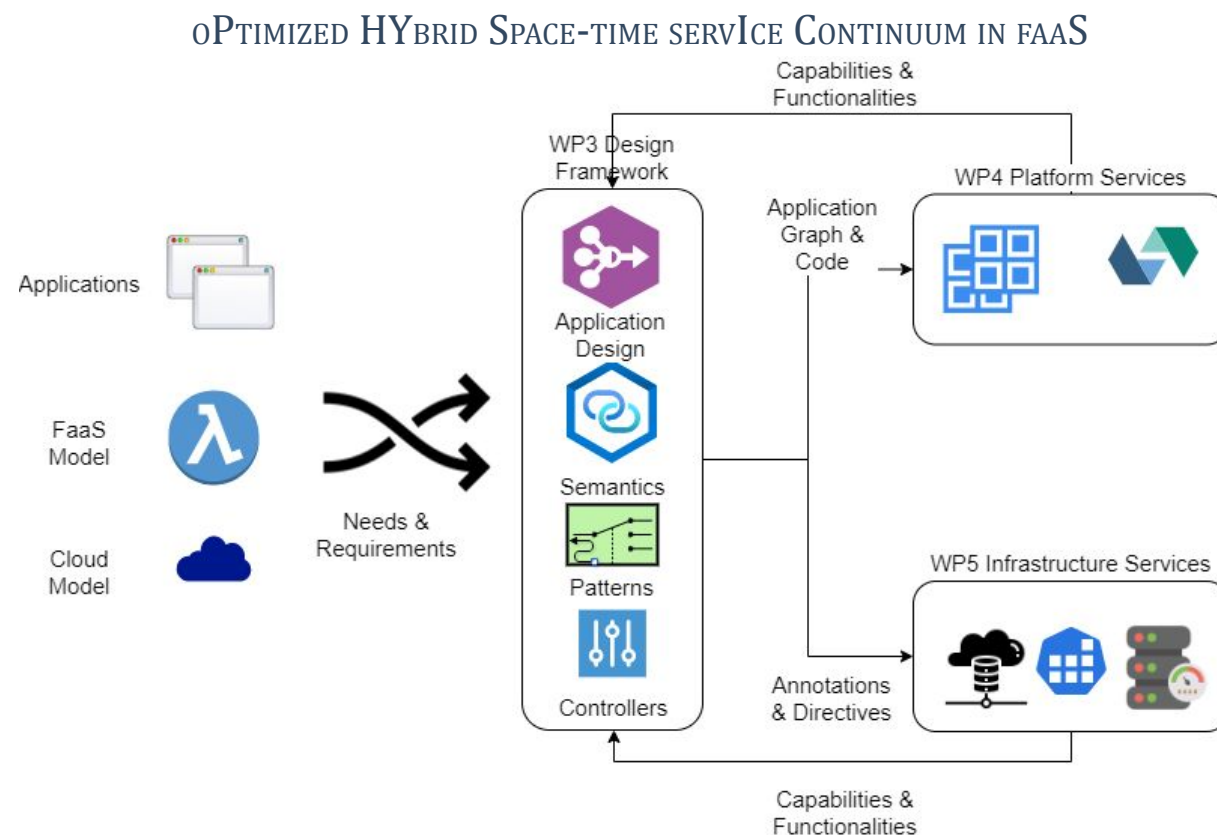


Project Goals

Visual programming environment to create serverless workflows with reusable patterns and increased semantics

Platform-level functionalities to orchestrate and deploy FaaS workflows and optimize cloud/edge interplay

Provider-local resource management mechanisms to offer competitive and optimized services execution





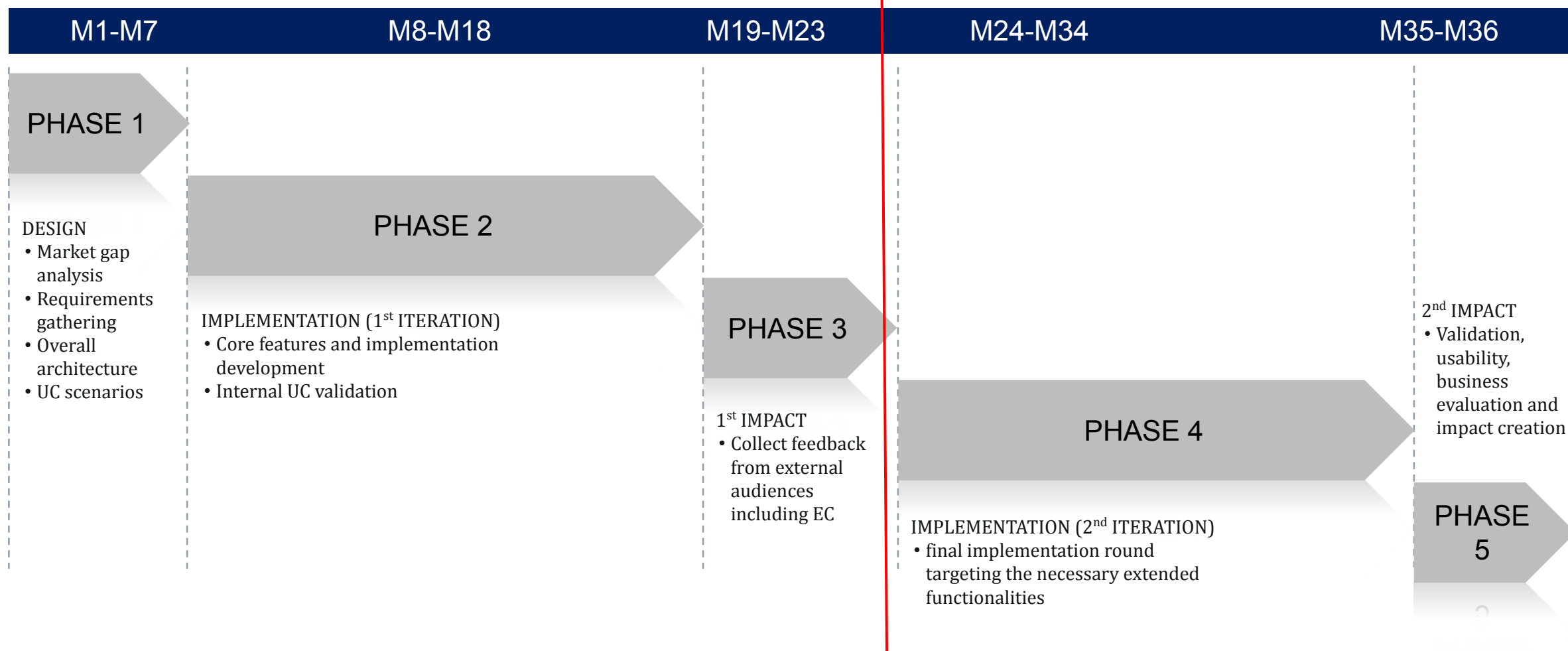
Project details

- ❖ H2020 ICT 40 - Cloud Computing: towards a smart cloud computing continuum
- ❖ Research & Innovation Action
- ❖ Project Budget: 4,985,712€
- ❖ EC Grant (100%): 4,985,712€
- ❖ Timeframe: 01-01-2021 to 31-12-2023
- ❖ 14 Partners
- ❖ GFT is Project Coordinator

	1. GFT – Italy		8. INQBIT – Romania
	2. ATOS IT – Spain		9. INNOVATION SPRINT – Belgium
	3. HPE – Italy		10. INNOV-ACTS – Cyprus
	4. RED HAT – Israel		11. CYBELETECH – France
	5. FUJITSU – Germany		12. UPM – Spain
	6. BYTE COMPUTER – Greece		13. HUA – Greece
	7. RYAX TECHNOLOGIES - France		14. DFKI - Germany



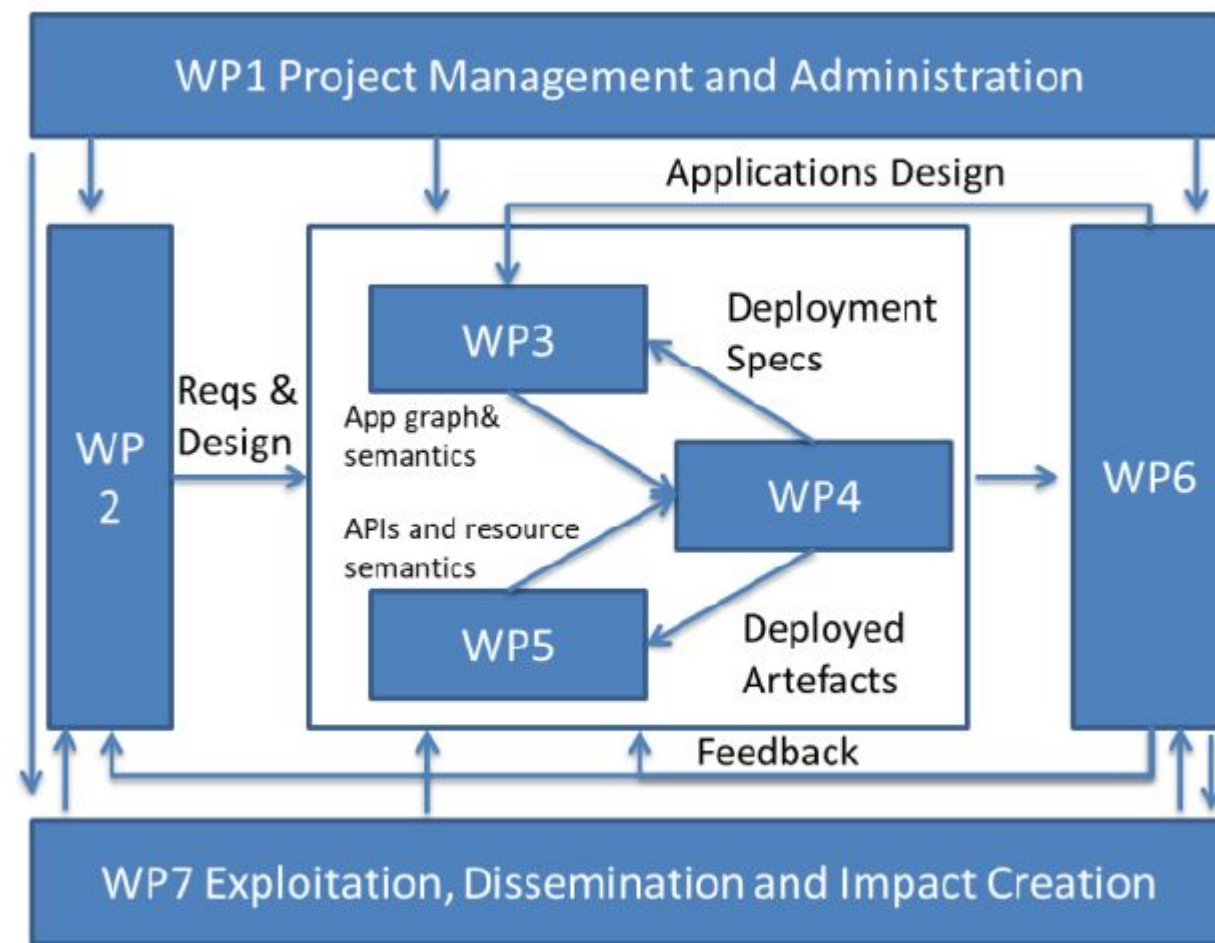
High Level Project Plan





WPs

- ❖ WP2: Requirements and Architecture
- ❖ WP3: User Layer
 - ❖ Design Environment & Function DevOps
 - ❖ Reusable Patterns, semantics and handles from WP4/5
- ❖ WP4: Platform Services
 - ❖ Semantics Management
 - ❖ Placement Optimization
 - ❖ Performance Analysis
 - ❖ Data Services
 - ❖ Deployment Orchestration Services
- ❖ WP5: Infrastructure Services
 - ❖ Resource Controllers & Semantics
 - ❖ Local placement optimizers (container node selection)
- ❖ WP6: Applications, Pilots and Platform Integration





PHYSICS Technical Architecture

WP3: User Layer

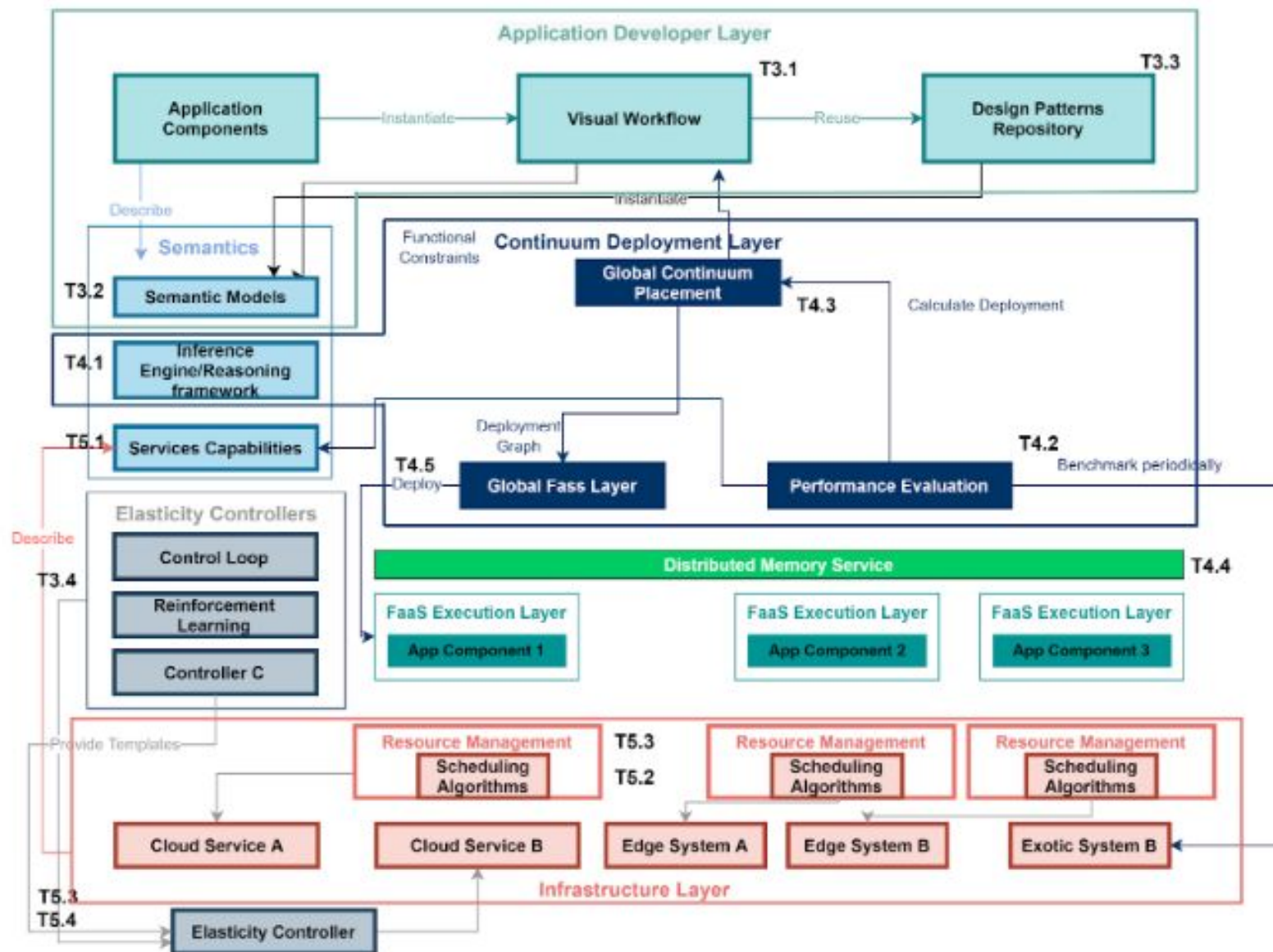
- ❖ Design Environment & Function DevOps
- ❖ Reusable Patterns, semantics and handles (e.g. controllers) from WP4/5

WP4: Platform Services

- ❖ Semantics Management
- ❖ Placement Optimization
- ❖ Performance Analysis
- ❖ Data Services
- ❖ Deployment Orchestration Services

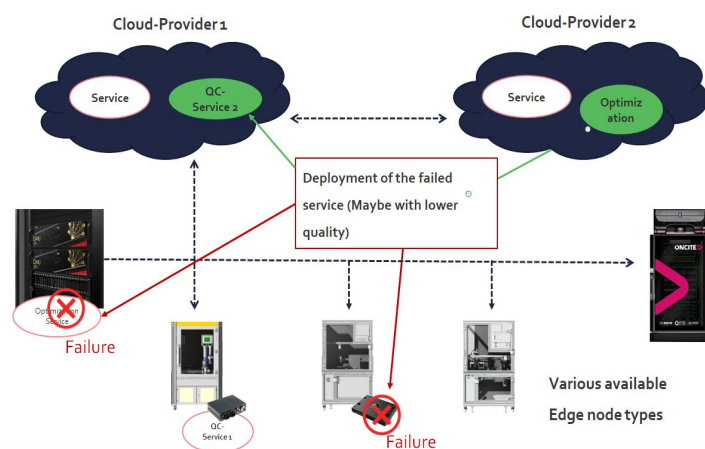
WP5: Infrastructure Services

- ❖ Resource Controllers & Semantics
- ❖ Local placement optimizers (container node selection)

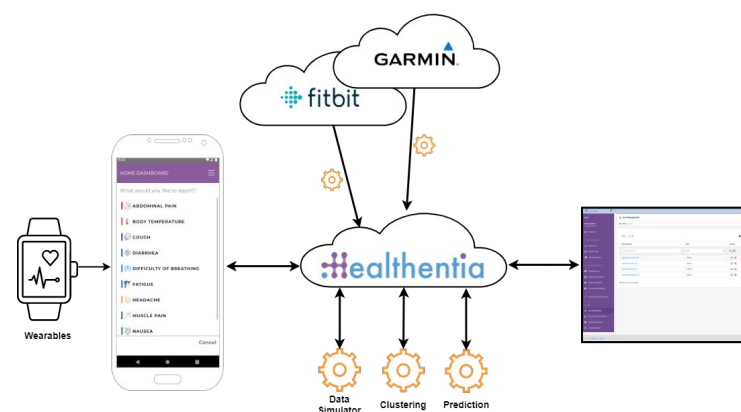


Project Pilots

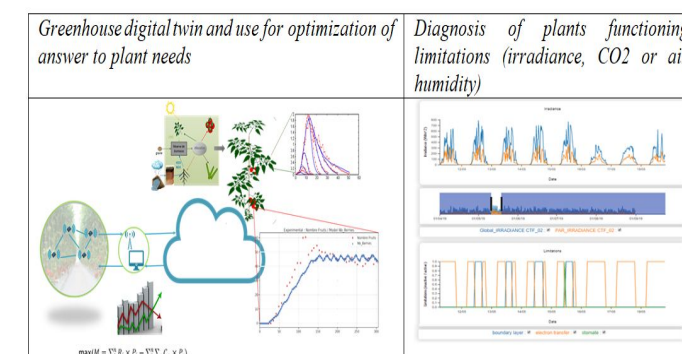
Smart Manufacturing for increased resilience and interplay



eHealth Personalized Monitoring and Collective Analysis



Smart Precision Agriculture





PHYSICS

PHYSICS Design Environment

Joint Work from GFT, HPE and HUA



This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047





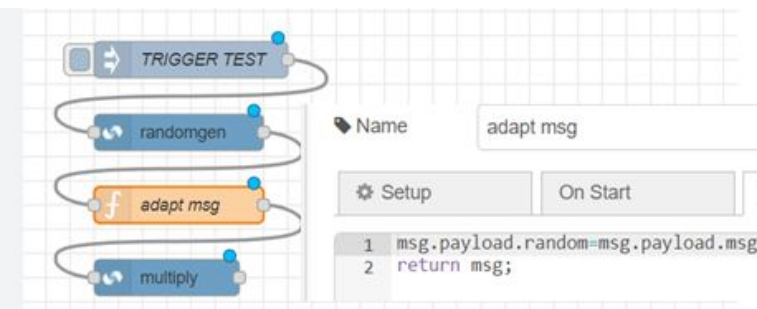
Goals of PHYSICS DE: Enhance, Abstract, Enrich FaaS application creation

- ❖ Customize function environment
- ❖ Simplify the creation of complex function workflows
 - ❖ Support complex primitives like Joins
- ❖ Exploit reusable patterns
- ❖ Increase semantics
- ❖ Abstract packaging, testing and deployment

191 Commits	13 Branches	0 Releases
Branch: george	test	New file Upload file HTTPS SSH https://repo.apps.ocphub.p
gkousiouris	6d25f2fb40	commit to deploy 3 days ago
data	6d25f2fb40	commit to deploy 3 days ago
.gitignore	718ca69f7	commit to deploy 8 months ago
Dockerfile	c3c13bf4fb	commit to deploy 3 weeks ago
Dockerfile_debian	718ca69f7	commit to deploy 8 months ago
Dockerfilecustom	60e6823a28	Updated to use flow file coming from MinIO 8 months ago
Jenkinsfile	e49fb2c3e5	commit to deploy 3 months ago
Jenkinsfile-base	07b8b88cce	add proper bibeline for building debian image 7 months ago
Jenkinsfile-ow	e49fb2c3e5	commit to deploy 3 months ago
README.md	7ec09907dc	commit to deploy 8 months ago
python-requirements.txt	718ca69f7	commit to deploy 8 months ago

```

- randomgen_function:
  call: http.get
  args:
    url: https://REGION/-PROJECT_ID/.cloudfunctions.net/randomgen
  result: randomgen_result
- multiply_function:
  call: http.post
  args:
    url: https://REGION/-PROJECT_ID/.cloudfunctions.net/multiply
    body:
      input: ${randomgen_result.body.random}
  result: multiply_result
- return_result:
  return: ${multiply_result}
    
```



Jenkins

Dashboard • TEST-multi • george • #113

- Back to Project
- Status
- Changes
- Console Output
- View as plain text
- Edit Build Information
- Parameters
- Timings
- Git Build Data
- Thread Dump

Console Output

Started by user Georgios Kousiouris

Posting JSON message to RabbitMQ:

```
{
  "url": "https://orchestrator.apps.ocphub.physics-faas.eu/job/TEST-multi/george/113/",
  "build_job_name": "TEST-multi/george",
  "build_number": 113,
  "jenkins_master_fqn": "jenkins-0",
  "state": "STARTED",
  "causes": "[\"UserInitCause, Started by user Georgios Kousiouris\"]",
  "parameters": {
    "deploy_devfalse": "false",
    "projectnameop3": "physics-faas",
    "flowfilehellfunctionV2_george_ebde738d-7ef9-4ed0-aaf4-4f2736afe395.json"
  }
}
```

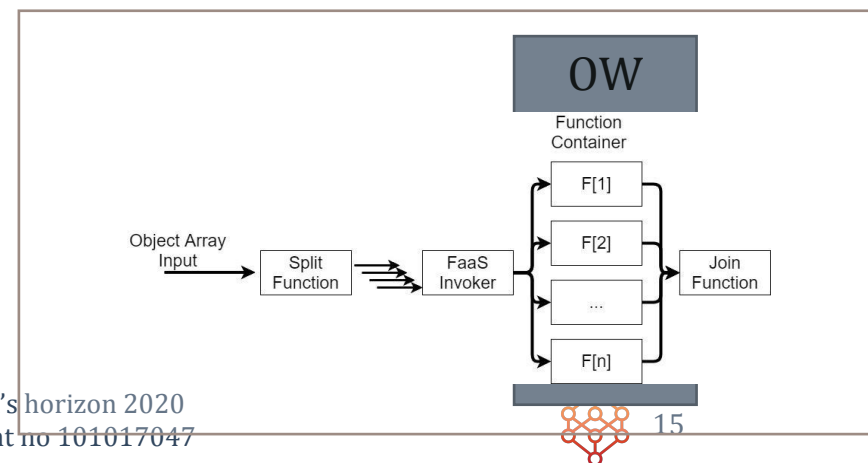
> git rev-parse --resolve-git-dir /var/jenkins_home/caches/git-091ca0d2438d4956b2c739c8a3b327/.git # timeout:10

Setting origin to https://repo.apps.ocphub.physics-faas.eu/PHYSICS/test.git

> git config remote.origin.url https://repo.apps.ocphub.physics-faas.eu/PHYSICS/test.git # timeout:10

Fetching origin...

Fetching upstream changes from origin



Union's horizon 2020

research and innovation programme under grant agreement no 101017047

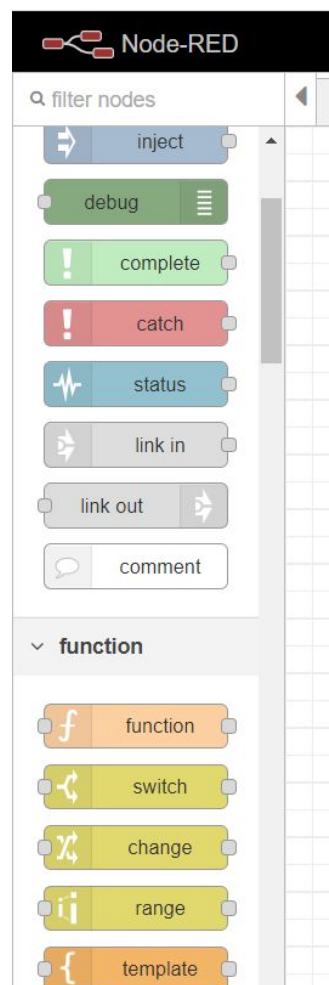
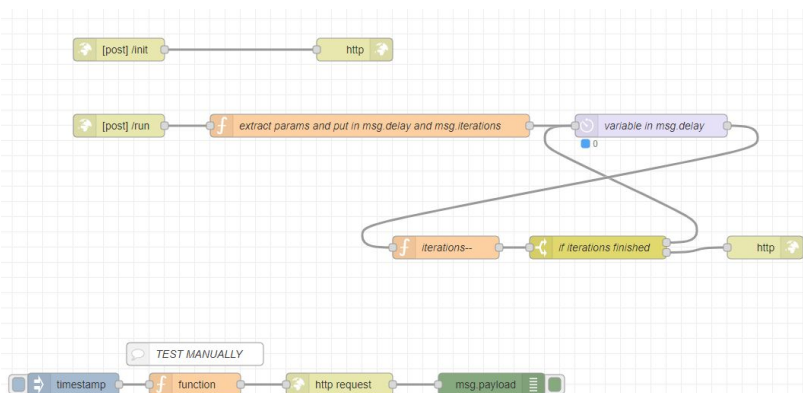




Baseline Technologies

❖ Node-RED

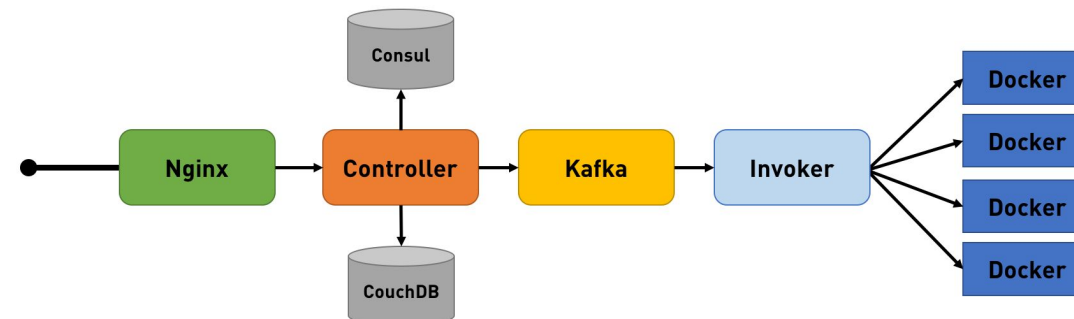
- ❖ Programming environment for event driven applications
 - ❖ Built-in nodes
 - ❖ NPM extension nodes
 - ❖ Subflows (groups of functions) as nodes
- ❖ Combined workflow orchestration and function execution abilities
- ❖ Used as the main execution **runtime** and function **choreographer** for PHYSICS functions



The research and innovation programme under grant agreement no 101017047

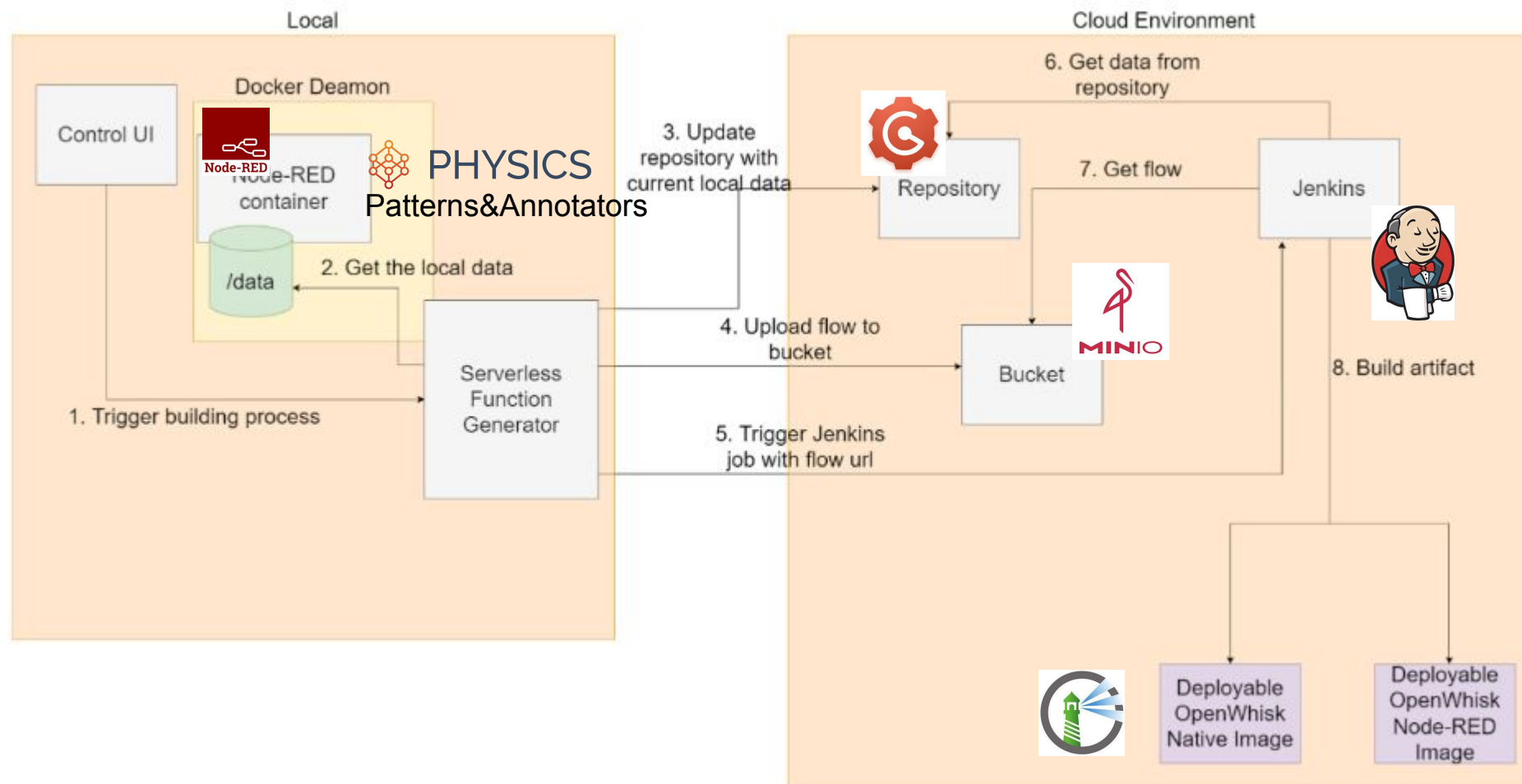
❖ Openwhisk

- ❖ Open source FaaS platform
- ❖ Supported by IBM
- ❖ Main tool behind IBM Cloud Functions Service
- ❖ Can execute functions based on custom container images



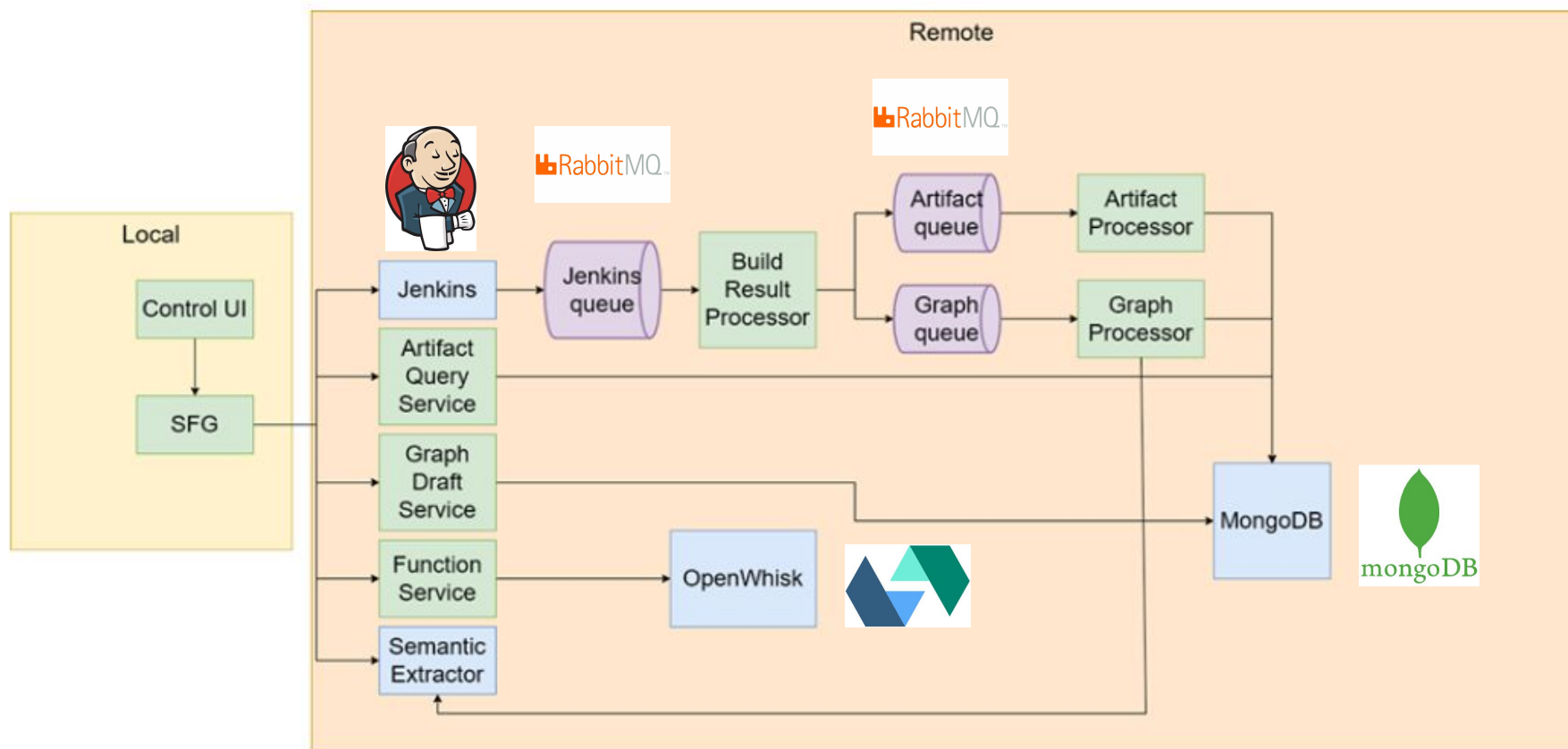


PHYSICS Design Environment Function Artefact Creation Lifecycle





PHYSICS Design Environment Backend





Design Environment Demo

❖ Full video available at: <https://www.youtube.com/watch?v=D02sFdfCD-o>

- ❖ Function creation
- ❖ Function Build Management
- ❖ Function Testing
- ❖ Logging
- ❖ App creation



PHYSICS

Admin Panel

Node Red

Branch
george

Logs



Admin Panel Builds Test Graphs



Available Flows:

HelloFunctionV2	39ef55a8.55f96a ✓	▼
Flow 4	a0c24f9aa45959ea	▼
TEST OW	682ea5c4e31c2450	▼
Sequence_v2	104348eeda17061b ✓	▼
Hello_Service_v2	7cf35d6fe22c0878 ✓	▼
Hello_DMS	3dc59d6aea6e35a4 ✓	▼
Test DMS+SJ	32b38369e8dc6e19	▼
splitjoinFunction	e98bf48d7a877d25 ✓	▼
Choreographer Service	d616400b75eac2fb ✓	▼
DYNAMIC CLUSTERS FOR LOCALITY	5f22488fe1374de3	▼



Embedded Node Red with enriched PHYSICS patterns and annotators- used to create and annotate flows

The screenshot displays the PHYSICS Node-RED interface. On the left sidebar, the 'Node Red' option is highlighted with a red box. The main workspace shows a flow diagram with nodes like '[post] /init', 'http', '[post] /run', and 'hello world'. A comment box says 'OW SKELETON with hello world function'. At the bottom right, a red box highlights the 'Executor Mode' and 'SizingAnnotator' buttons.





View and build list of available functions, created in Node Red tool.



PHYSICS

Admin Panel

Node Red

Branch
master



Logs



Test

bb5bb4964ce8bd2a ✓



New test flow

144b41aec7c48ed8 ✓



Document ID	URL	Action name	Built date	
62b972a9f245128e087ca33f	registry.apps.ocphub.physics-faas.eu/custom/master:181	New_test_flow_master_0b0db467-a484-4023-9acc-626931e3afe5.json	6/27/22, 11:04 AM	
62b973d4f245128e087ca340	registry.apps.ocphub.physics-faas.eu/custom/master:182	New_test_flow_master_84f3a58e-f092-4697-8981-c5fe490d2f54.json	6/27/22, 11:09 AM	

Rebuild

Next flow

3dcaccf50e4be70d ✓



HelloFunctionV2

506cf45e1ab62db5 ✓





Invoke OpenWhisk functions built by user or for other branches

Admin Panel

Node Red

Branch master

Logs

Function *

test

My branch

Test_master_17be2c66-9985-4f47-af47-eb2dcb176124.json

Test_master_8af985c9-be02-43ec-a168-2d11513343d3.json

Test_master_ae3343a9-eb2e-4945-af87-9e60f674fdc2.json

Test_master_d50e4333-667b-4b5e-9c4c-d17aa9102538.json

Other

HelloSPEC_testNode_george_dcacbf41-e154-42e6-a632-58bb8f20af44.json

HelloSPEC_testNode_george_d01e7b5e-d116-4657-b90c-1e578a6c93f5.json

HelloSPEC_testNode_george_8dbe280d-ed7b-4ee6-a3f2-373375973965.json

☒ Check automatically until success

Setting parameters for invoke

Put JSON for parameters

Parameter key *	Parameter value *
first	test value
Parameter key *	Parameter value *
second	other value
+	

Invoke





Example view for the result of function invocation.

Invoke

☒ Check automatically until success - requests: 6, updated at 10:39:47 AM

Check result

Activation ID: 070f49fc6ed74da48f49fc6ed72da488

Most recent results

Hello_DMS_george_a763502c-95ae-4f88-91d3-047a2b02d4ba.json

070f49fc6ed74da48f49fc6ed72da488

Params

name	Blazej
------	--------

Result

```
{
  "data": "hello Blazej",
  "dmsResult": {
    "error": "The action did not return a dictionary."
  }
}
```





App creation from selected graphs: package and manage multiple functions as one application for deployment

Admin Panel

Node Red

Branch
master

Logs

Create Graph

Name:

Flows

Flow 1	https://registry.apps.ocphub.physics-faas.eu/custom/master:214
Flow 2	registry.apps.ocphub.physics-faas.eu/custom/master:200
Hello_DMS	registry.apps.ocphub.physics-faas.eu/custom/master:190
Test DMS+SJ	https://registry.apps.ocphub.physics-faas.eu/custom/master:193
splitjoinFunction	registry.apps.ocphub.physics-faas.eu/custom/master:203
Choreographer Service	https://registry.apps.ocphub.physics-faas.eu/custom/master:184

App Graph

Flow 4	registry.apps.ocphub.physics-faas.eu/custom/master:187
TEST OW	registry.apps.ocphub.physics-faas.eu/custom/master:188

Cancel Create



Created Draft view

Consists of the flows that are already built

and

Flows for which had been triggered built process.

Draft 15

Built flows

- ✓ registry.apps.ocphub.physics-faas.eu/custom/george:97
- ✓ registry.apps.ocphub.physics-faas.eu/custom/george:98

Flows waiting for build

- 🕒 Choreographer Service
d616400b75eac2fb
- 🕒 splitjoinFunction
e98bf48d7a877d25



Displaying logs from microservices with useful information

Diagram illustrating the structure of log entries from microservices, with labels pointing to specific fields:

- timestamp
- microservice name
- source class
- client ID
- message

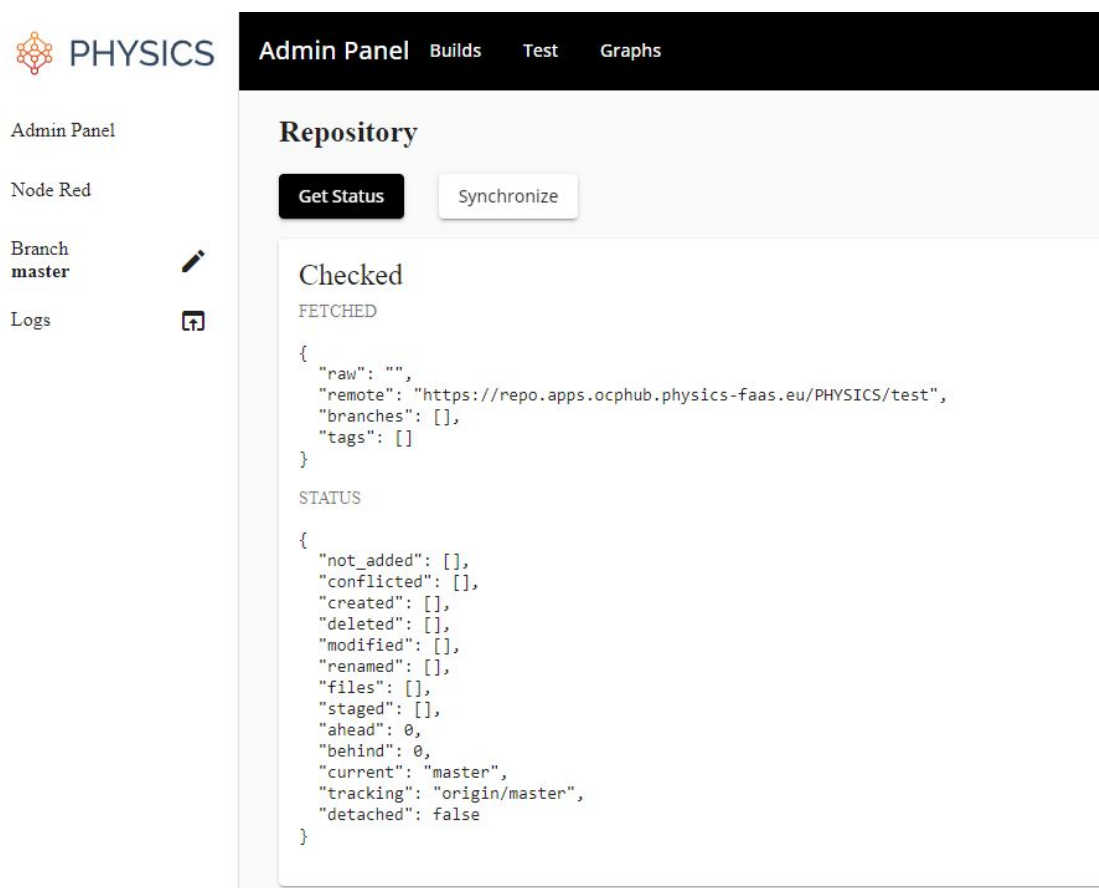
```
12:26:07.274 sfg ArtifactService user-id Getting artifact for 19 flows from http://127.0.0.1:3002/api/artifact
12:26:07.291 artifact-query-service ArtifactController user-id Get artifacts from the service: 19
12:26:07.293 artifact-query-service ArtifactService user-id Select artifacts from DB: admin.artifacts
12:26:07.727 sfg ArtifactService user-id Artifacts to return: 189
12:31:49.029 function-service FunctionService user-id Got list of functions, found results: 30
12:31:55.945 function-service FunctionController user-id Invoke function Open
```

Optional payload as JSON with button for easier copying

```
{
  "functionName": "Test_master_17be2c66-9985-4f47-af47-eb2dcb176124.json",
  "params": {
    "name": "Blazej"
  },
  "result": {
    "activationId": "99d2ef472723483692ef4727237836e8"
  }
}
```

User repository synchronization tool

View for checking the status



The screenshot shows the PHYSICS Admin Panel with a sidebar containing links to Admin Panel, Node Red, Branch master, and Logs. The main content area is titled 'Repository' and includes 'Get Status' and 'Synchronize' buttons. Below these buttons, the 'Checked' status is displayed as 'FETCHED'. A JSON object shows repository details: raw URL, remote URL (https://repo.apps.ocphub.physics-faas.eu/PHYSICS/test), branches, and tags. The 'STATUS' section shows a detailed JSON object with fields like not_added, conflicted, created, deleted, modified, renamed, files, staged, ahead, behind, current, tracking, and detached.

```
{
  "raw": "",
  "remote": "https://repo.apps.ocphub.physics-faas.eu/PHYSICS/test",
  "branches": [],
  "tags": []
}
```

```
{
  "not_added": [],
  "conflicted": [],
  "created": [],
  "deleted": [],
  "modified": [],
  "renamed": [],
  "files": [],
  "staged": [],
  "ahead": 0,
  "behind": 0,
  "current": "master",
  "tracking": "origin/master",
  "detached": false
}
```

Result of the synchronization of the repository



The screenshot shows the 'Repository' section with 'Get Status' and 'Synchronize' buttons. Below the buttons, the 'Result of synchronization' is displayed as 'PULLED'. A JSON object shows the result of the synchronization, including remoteMessages, created, deleted, files, deletions, insertions, and a summary of changes, deletions, and insertions.

```
{
  "remoteMessages": {
    "all": []
  },
  "created": [],
  "deleted": [],
  "files": [],
  "deletions": {},
  "insertions": {},
  "summary": {
    "changes": 0,
    "deletions": 0,
    "insertions": 0
  }
}
```




Deployment status

Button for starting
the deployment

app1

Flows

 Sequence_v2
<https://registry.apps.ocphub.physics-faas.eu/custom:18>

Deploy

app2



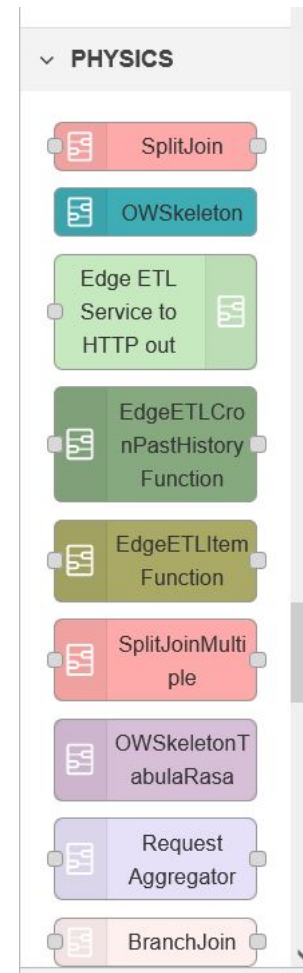
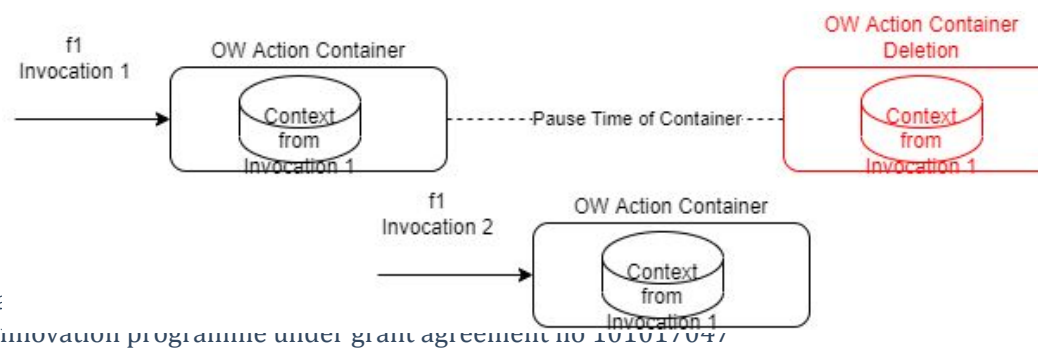
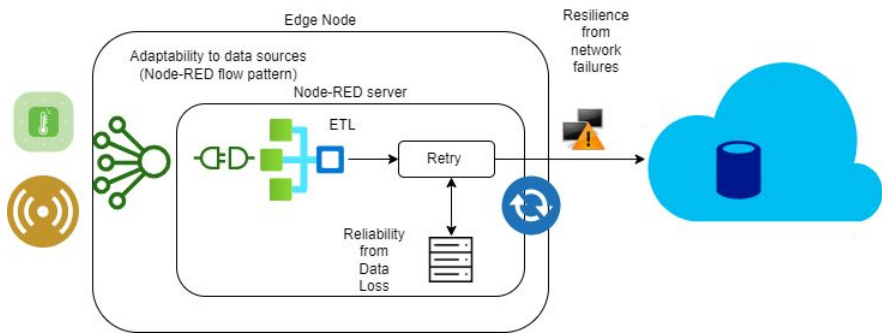
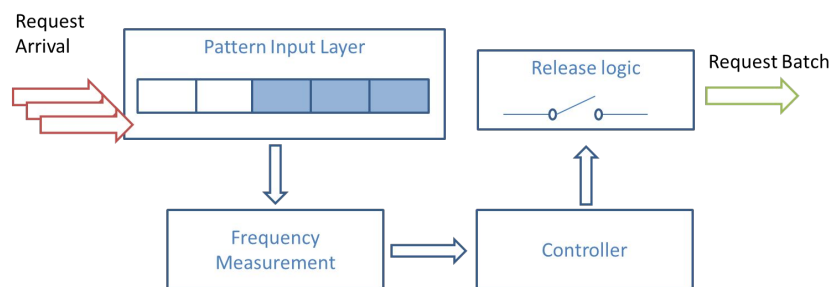
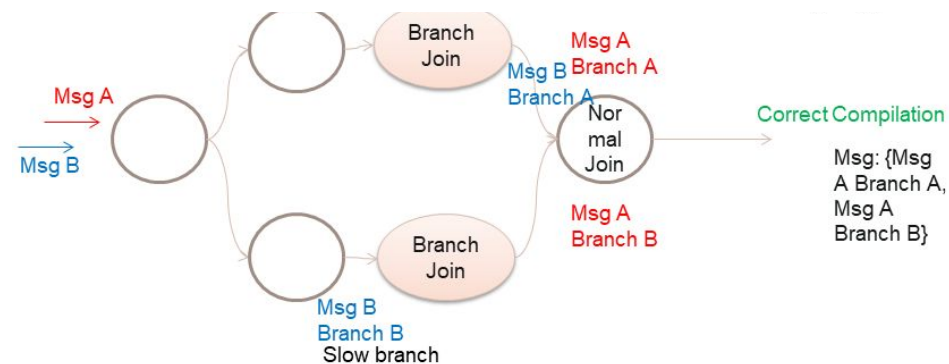
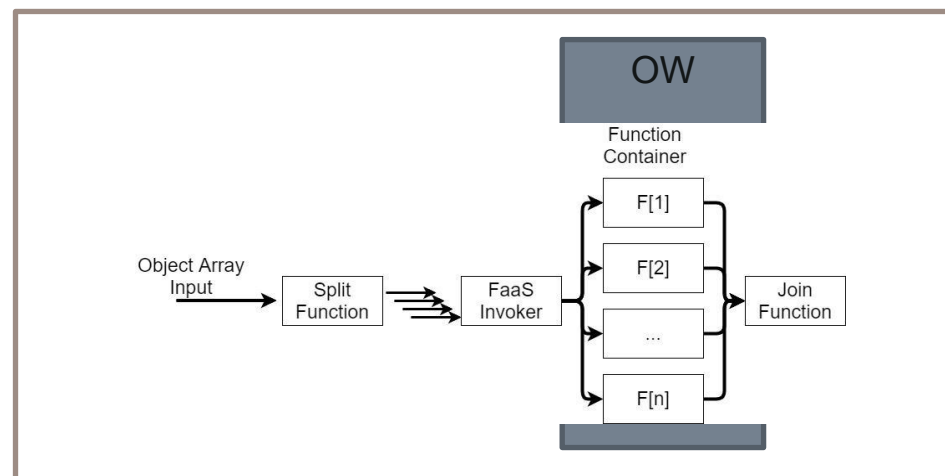


PHYSICS

Patterns to support reusability, manageability and abstracted functionality

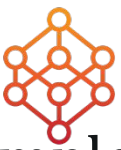
- ❖ Parallelization
- ❖ Context Management
- ❖ Data Collection
- ❖ Request Management
- ❖ Workflow primitives
- ❖ Etc.

OPTIMIZED HYBRID SPACE-TIME SERVICE CONTINUUM IN FAAS



This project has received funding from the European Union's Horizon research and innovation programme under grant agreement No 101017047

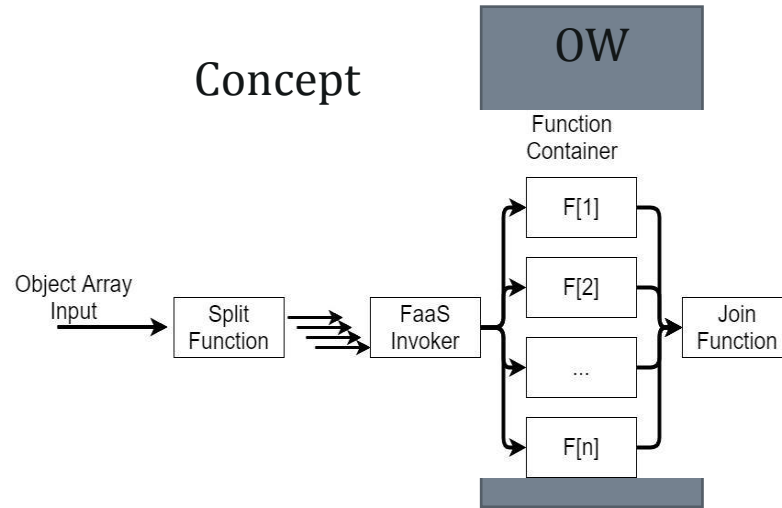




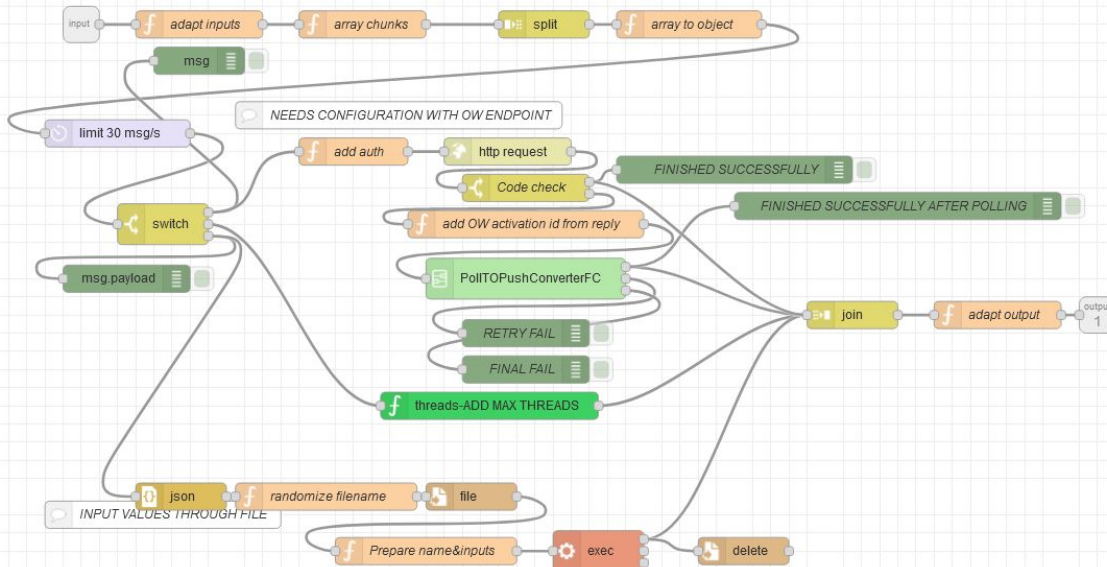
PHYSICS

Example of Fork-Join parallelization pattern

Concept

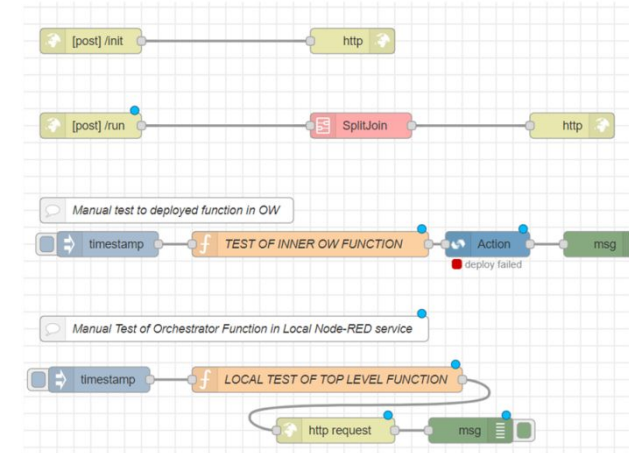


Implementation

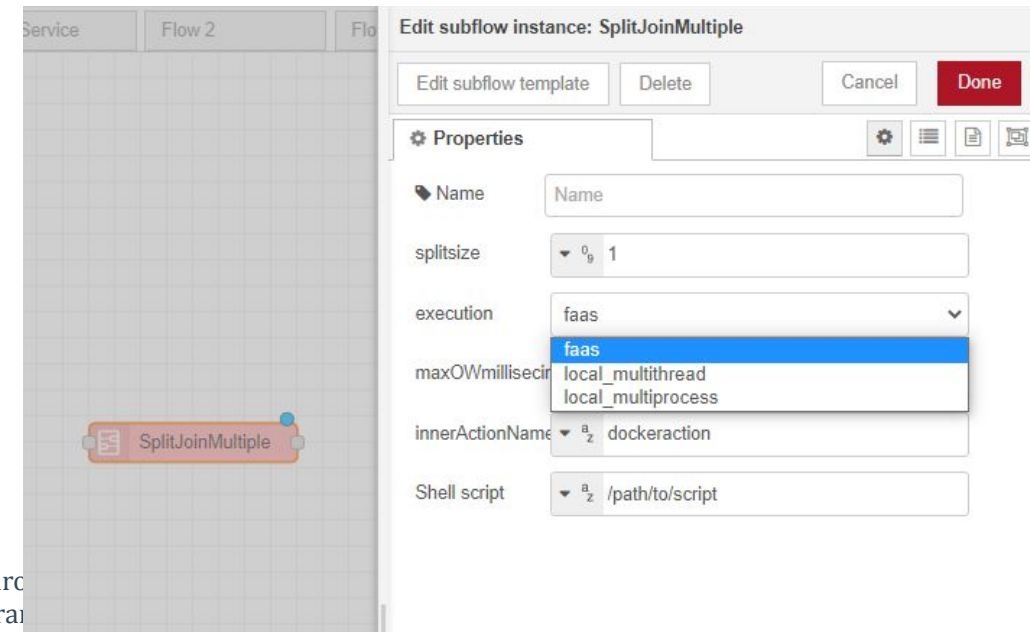


OPTIMIZED HYBRID SPACE-TIME SERVICE CONTINUUM IN FAAS

Usage in a Function



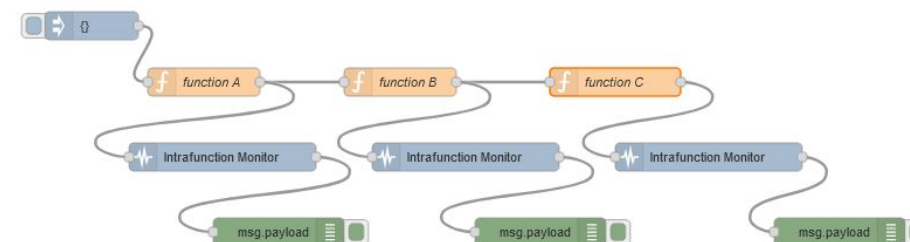
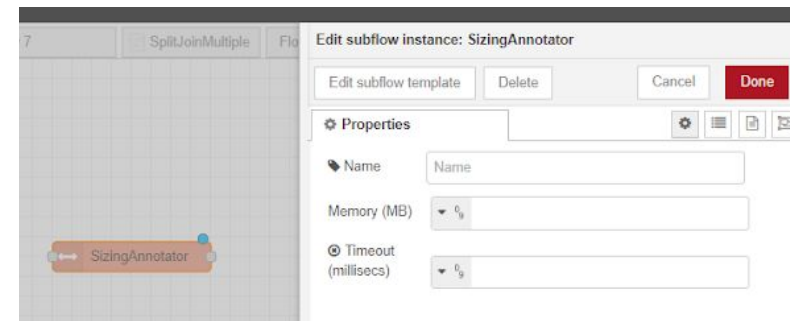
Parameterization



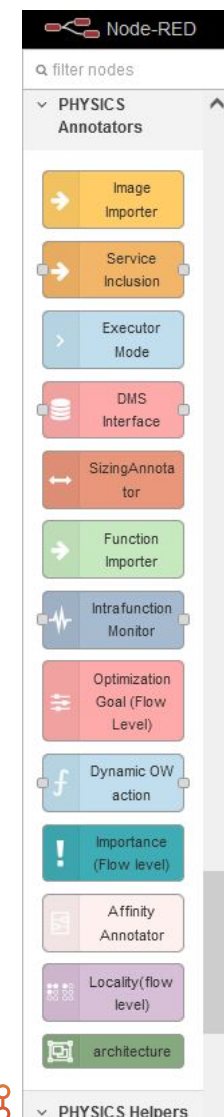


Available semantic nodes

- ❖ Function relative placement (Affinity, Antiaffinity)
- ❖ Function Locality (e.g. Cloud/Edge)
 - ❖ Dynamically retrieves available clusters
- ❖ Importance (High/Medium/Low)
- ❖ OptimizationGoal (Performance, Energy etc)
- ❖ Function Sizing
- ❖ CPU Architecture requirements
- ❖ Intrafunction monitoring
- ❖ DMS interface for linking to data services
 - ❖ Can indicate data locality



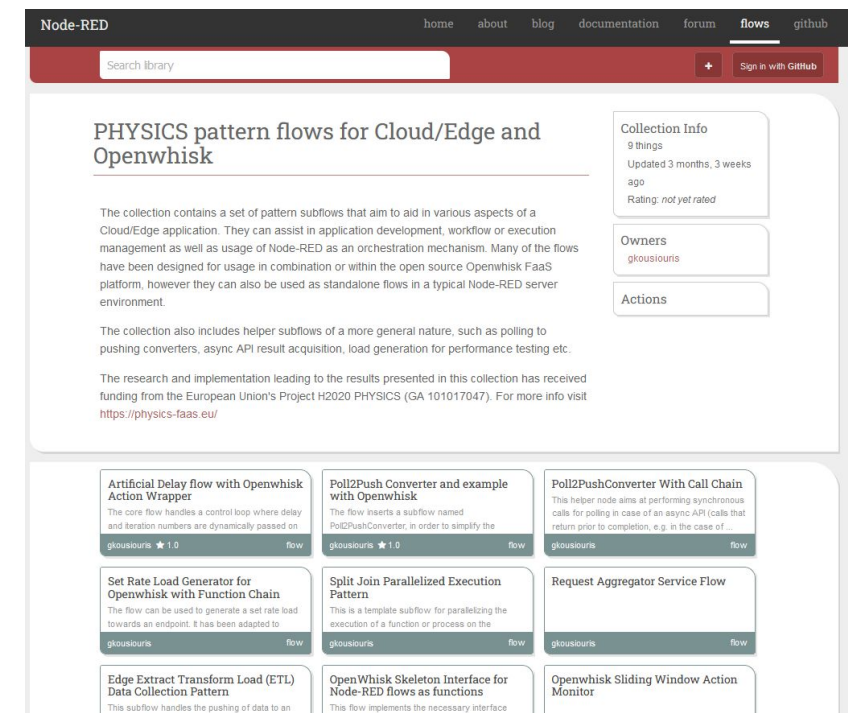
```
"consumesREST": [ + ],
"definesInterface": [ + ],
"dms_job_id": "ss",
"dms_location": "https://openwhisk.apps.ocphub.physics-faas.eu/api/v1",
"dms_namespace": "wp4namespace",
"dms_operation": "dms-w",
"dms_pocket_dst_file": "dd",
"dms_size": "10",
"executorMode": "NoderedFunction",
```



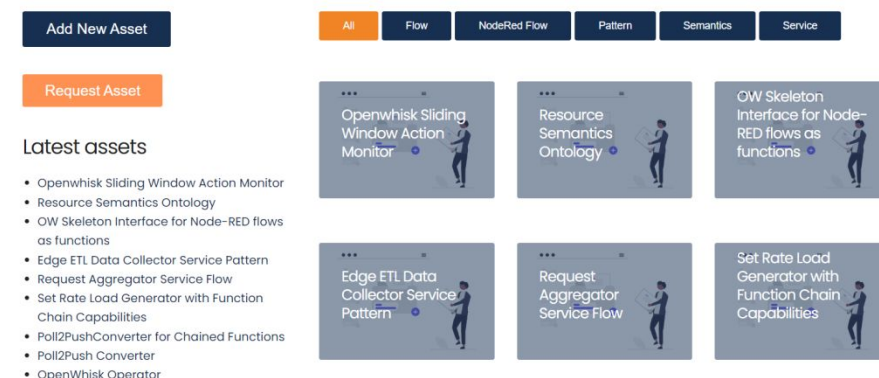


Node-RED specific artefacts

- ❖ Pattern implementations as Node-RED reusable subflows
 - ❖ Exploitable as individual artefacts
 - ❖ Inclusion in Node-RED repo with documentation
 - ❖ <https://flows.nodered.org/collection/HXSkA2IJLcGA>
 - ❖ Included in PHYSICS RAMP as artefacts along with other PHYSICS components
 - ❖ <https://marketplace.physics-faas.eu/assets>



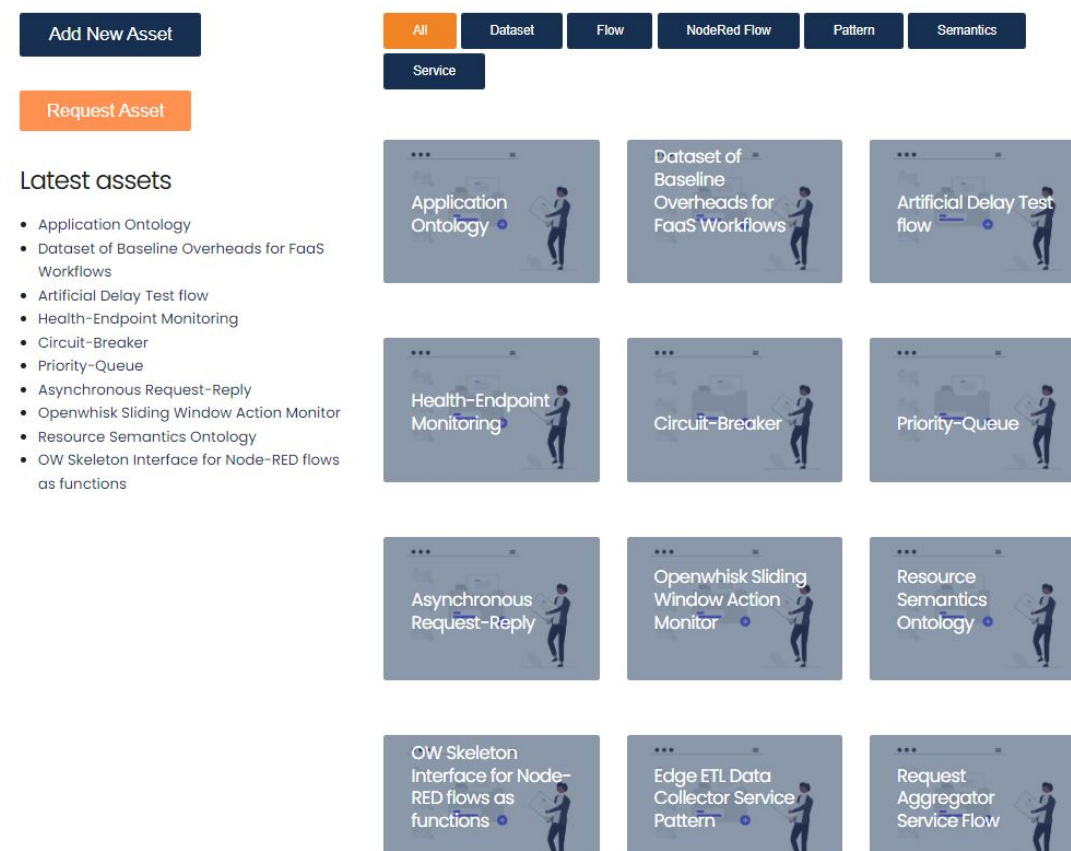
View Tutorial





PHYSICS RAMP

- ❖ Reusable Artefacts Marketplace Platform
 - ❖ <https://marketplace.physics-faas.eu/assets>
- ❖ Catalogue of available PHYSICS artefacts
 - ❖ Operators, Node-RED flows, Datasets, Services, Semantic Models
 - ❖ Links to repositories and usage guides
 - ❖ Ability to download and use
 - ❖ **Looking for testers from the community!**
 - ❖ Ability to request a specific asset or an extension to an asset





PHYSICS

PHYSICS Performance Aspects

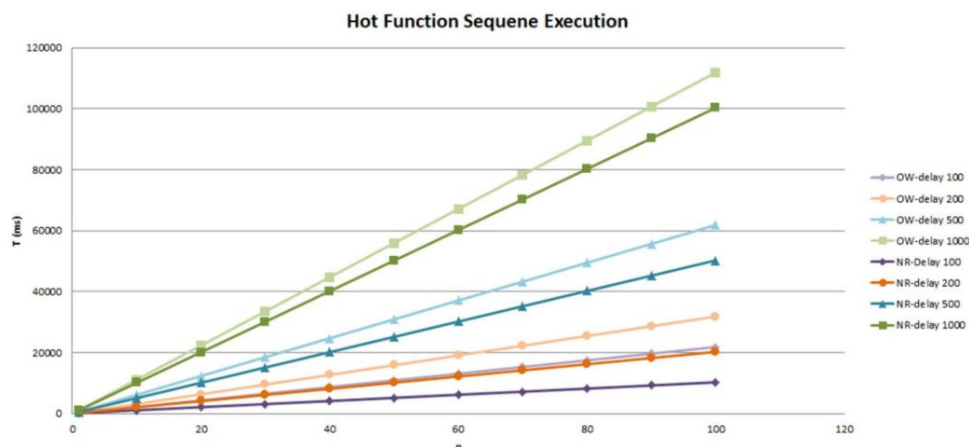


This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

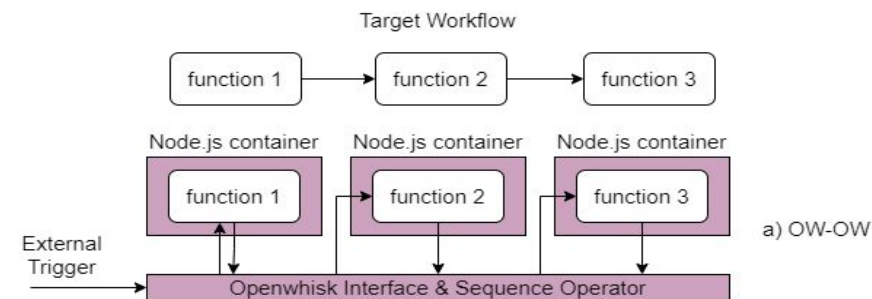




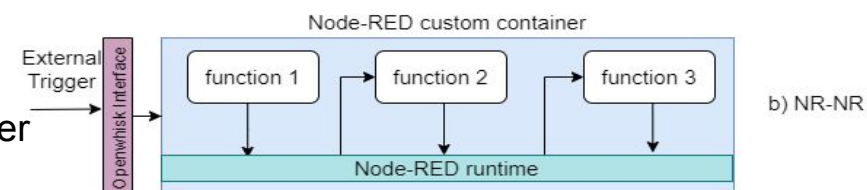
Function Choreography Overheads in Different Modes



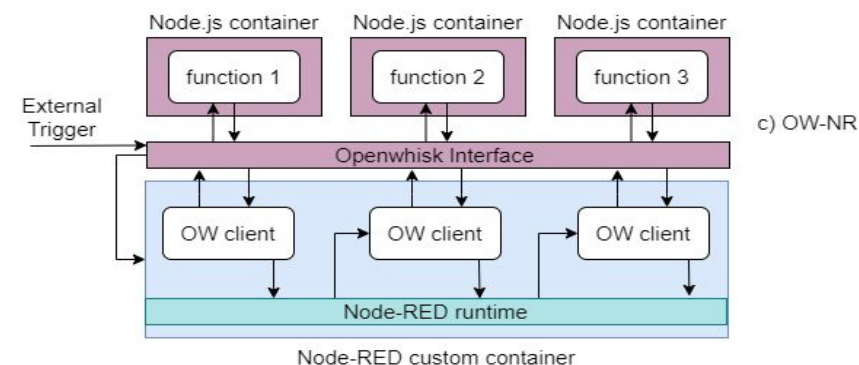
Sequences Only



Full
primitives/Intra-container
Parallelization



Full primitives/Full
parallelization



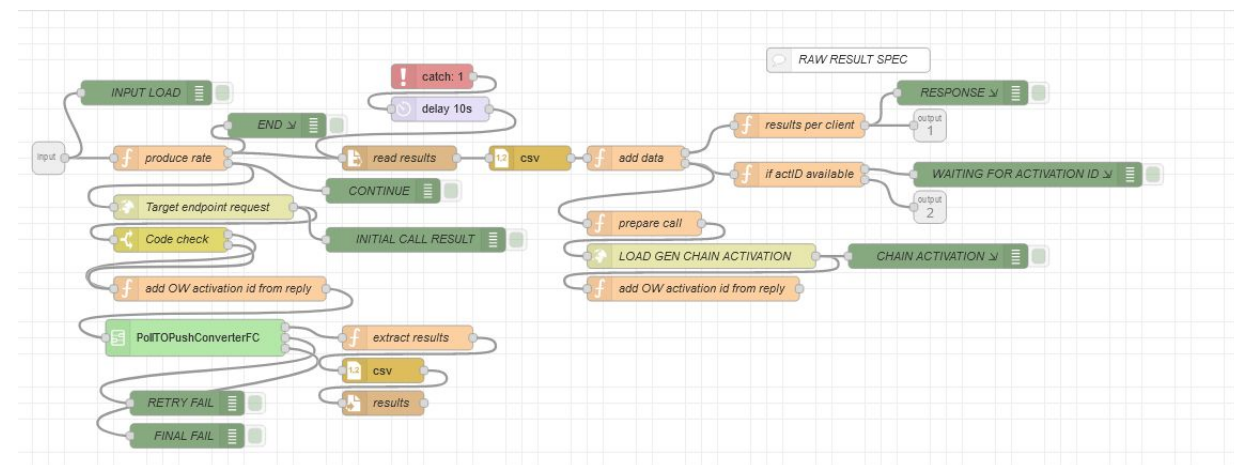
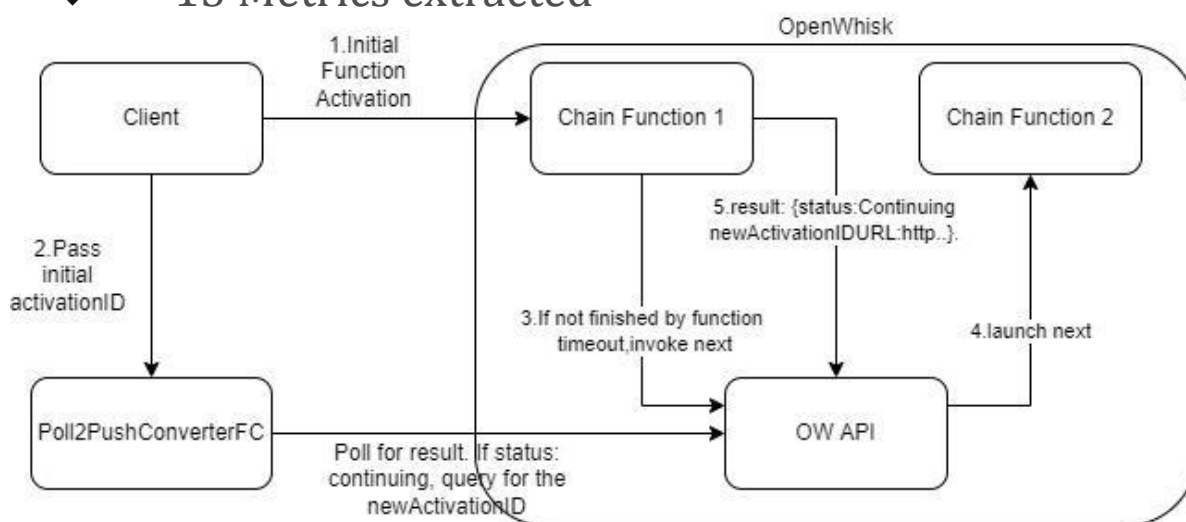
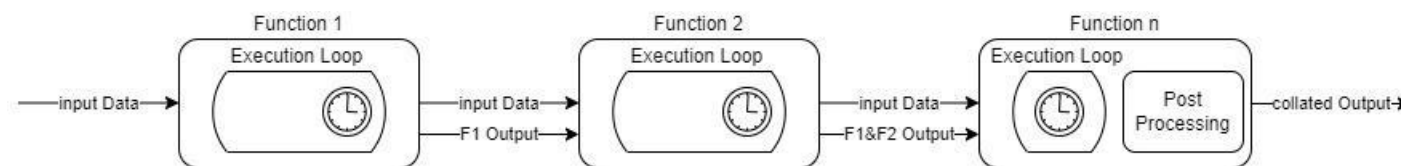
Mode	Warm Function Sequence Execution (ms)	Cold (ms)	Penalty
OW-OW	$T_{total} = 2.11 + n \times (T_{functiondelay} + 118.94)$	+2248	
NR-NR	$T_{total} = 128.36 + n \times (T_{functiondelay} + 1.69)$	+7373	
OW-NR	$T_{total} = 114.08 + n \times (T_{functiondelay} + 121.70)$	+9621	





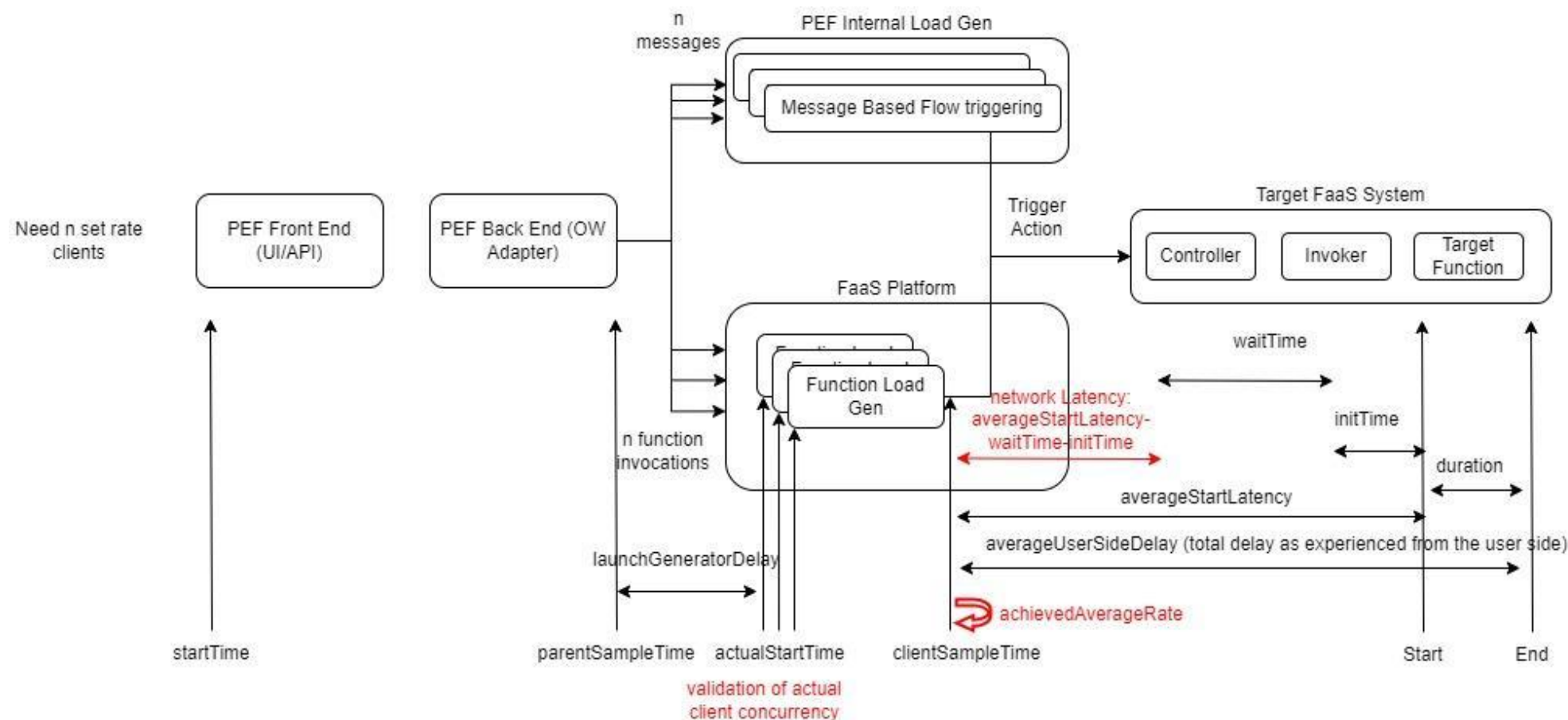
PHYSICS Set Rate Load Generator

- ❖ Node-RED subflow
 - ❖ Can be executed as a function
- ❖ Supports function chaining
 - ❖ To bypass OW timeout
- ❖ ~15 Metrics extracted





PHYSICS Load Generator Metrics

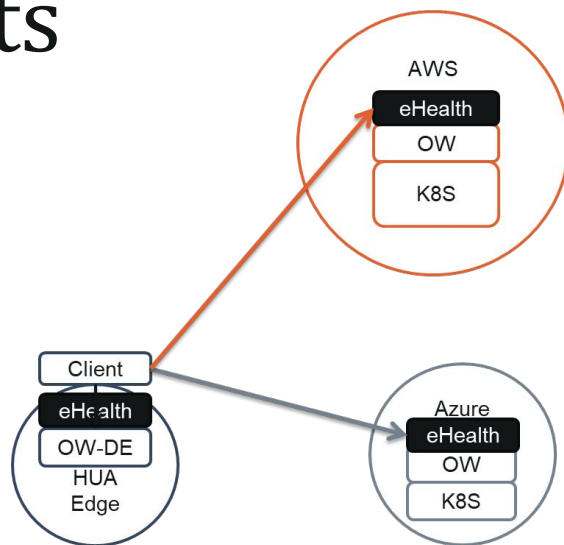




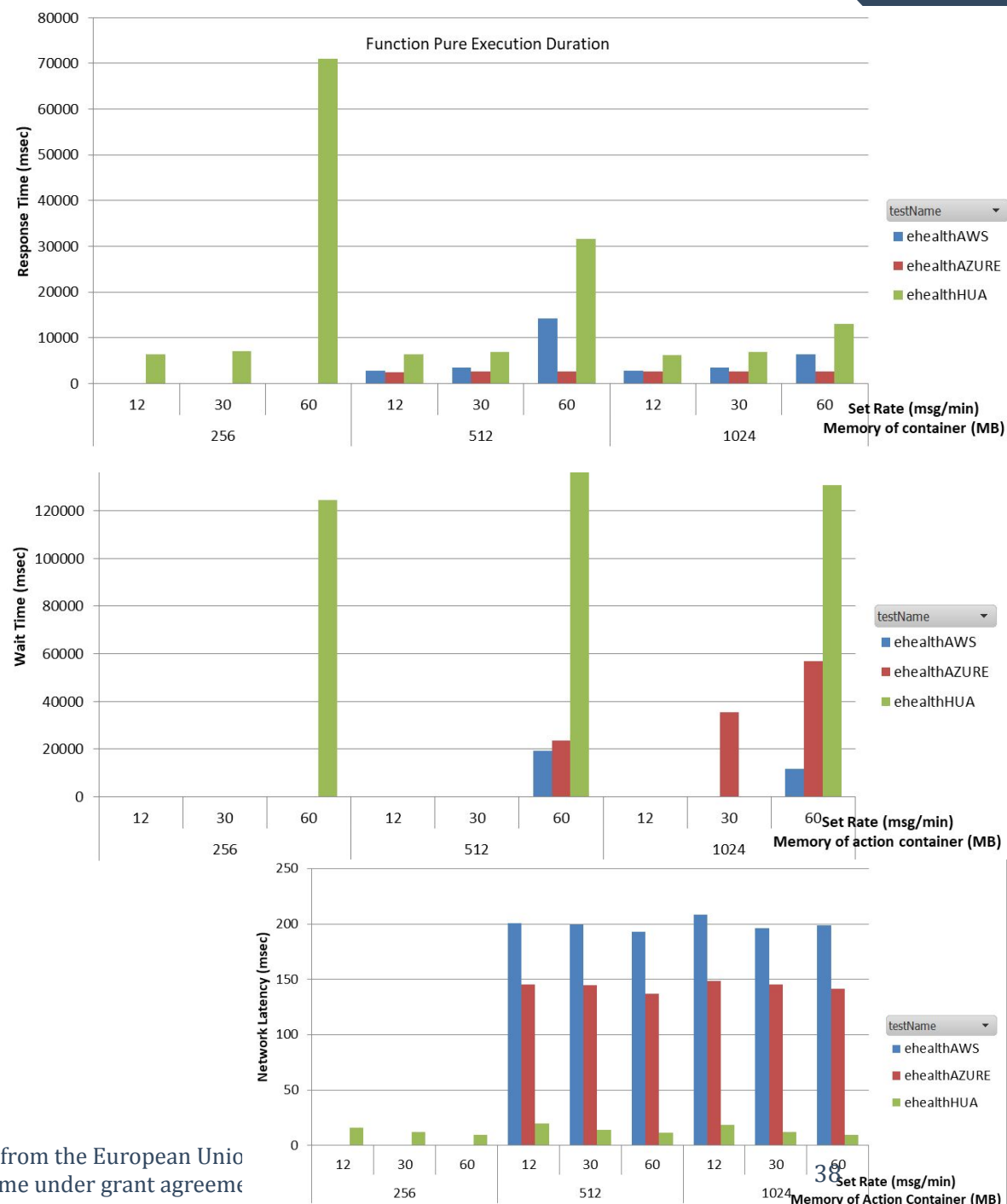
PHYSICS

Space time continuum measurements

- ❖ Identification of
 - ❖ Function sizing considerations
 - ❖ Main cluster features
 - ❖ As interpreted from relative execution time, wait time, latency
 - ❖ Missconfigurations
 - ❖ Concurrent containers effect



Cluster	#workers	Worker Size	Container Pool Memory per Node	Location
HUA	1	4 vCPU-8 GB RAM	8192 MB	Greece
AWS	4	4 vCPU-16 GB RAM	8192 MB	Sweden
Azure	3	8 vCPU-32 GB RAM	2048 MB	Netherlands





PHYSICS

RED HAT ROLE AND MAIN TECHNOLOGIES



This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047



Red Hat Main Role

❖ Infrastructure

❖ Lower layers of the stack

❖ Selection of infrastructure components: K8s, multicluster orchestration & networking

❖ Infrastructure and APIs extensions

❖ Integration

❖ API definitions (CRDs)

❖ Kubernetes Operators

❖ Kubernetes Webhooks



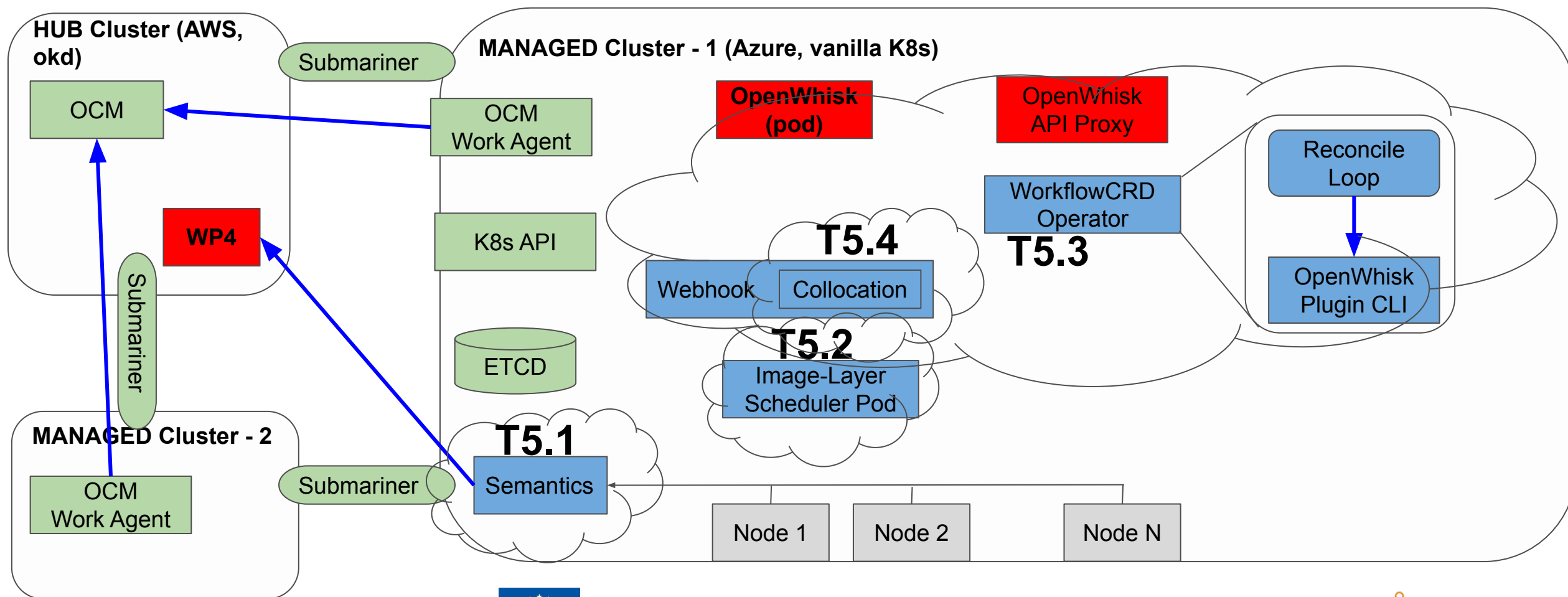
Technologies

- ❖ Kubernetes as the core (also OKD, OpenShift Community version)
- ❖ Submariner and Open Cluster Management (OCM) for multicluster setup
- ❖ MicroShift for low-footprint k8s devices
- ❖ Prometheus for monitoring data (both infrastructure and applications)
- ❖ OpenWhisk for Function as a Service
- ❖ K8s Operators and Webhooks
 - ❖ Kubernetes way to manage applications/components running on top of Kubernetes



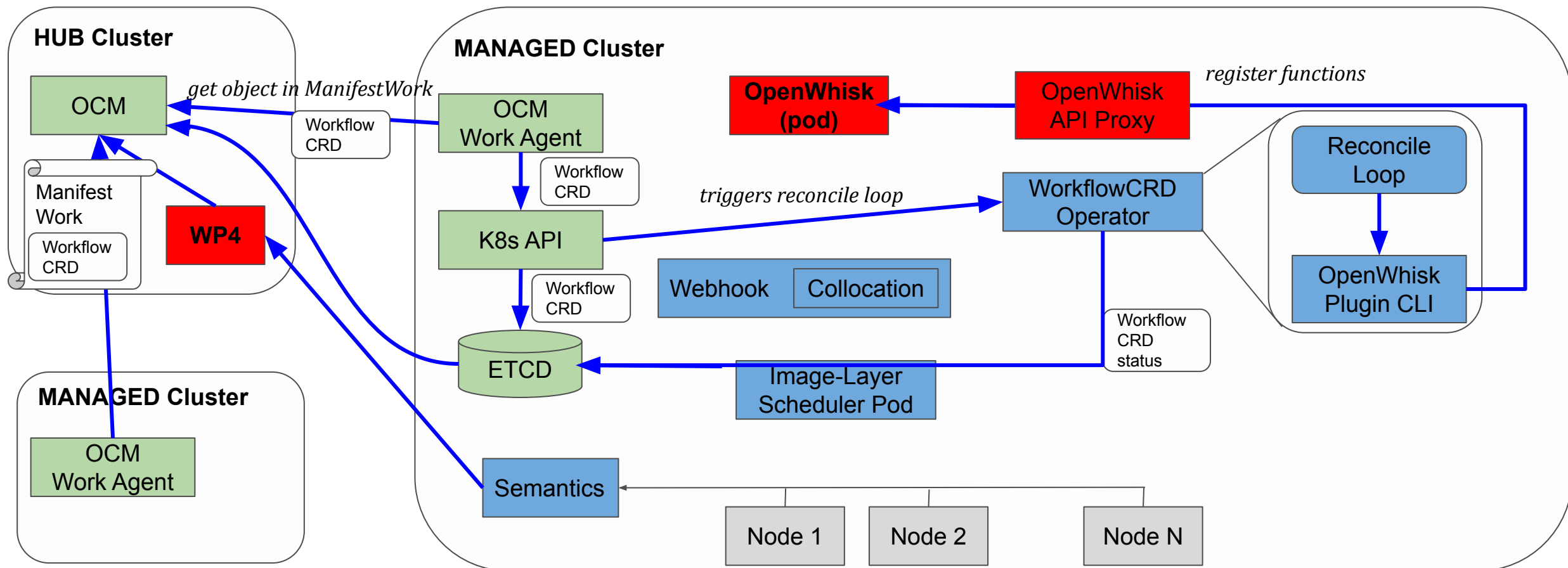


Components architecture



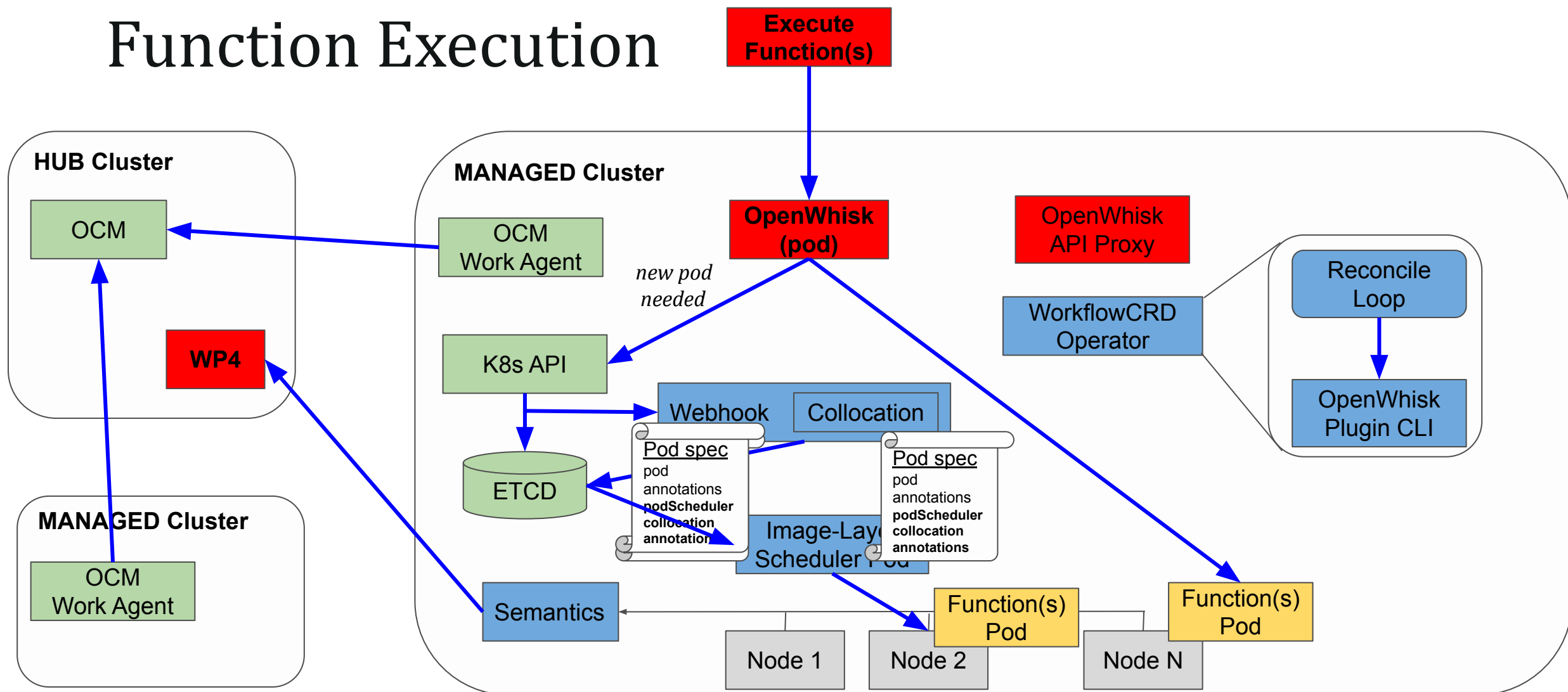


Function Registration





Function Execution





PHYSICS

K8s IMAGE LAYER SCHEDULING: BENEFITS OF OPEN SOURCE



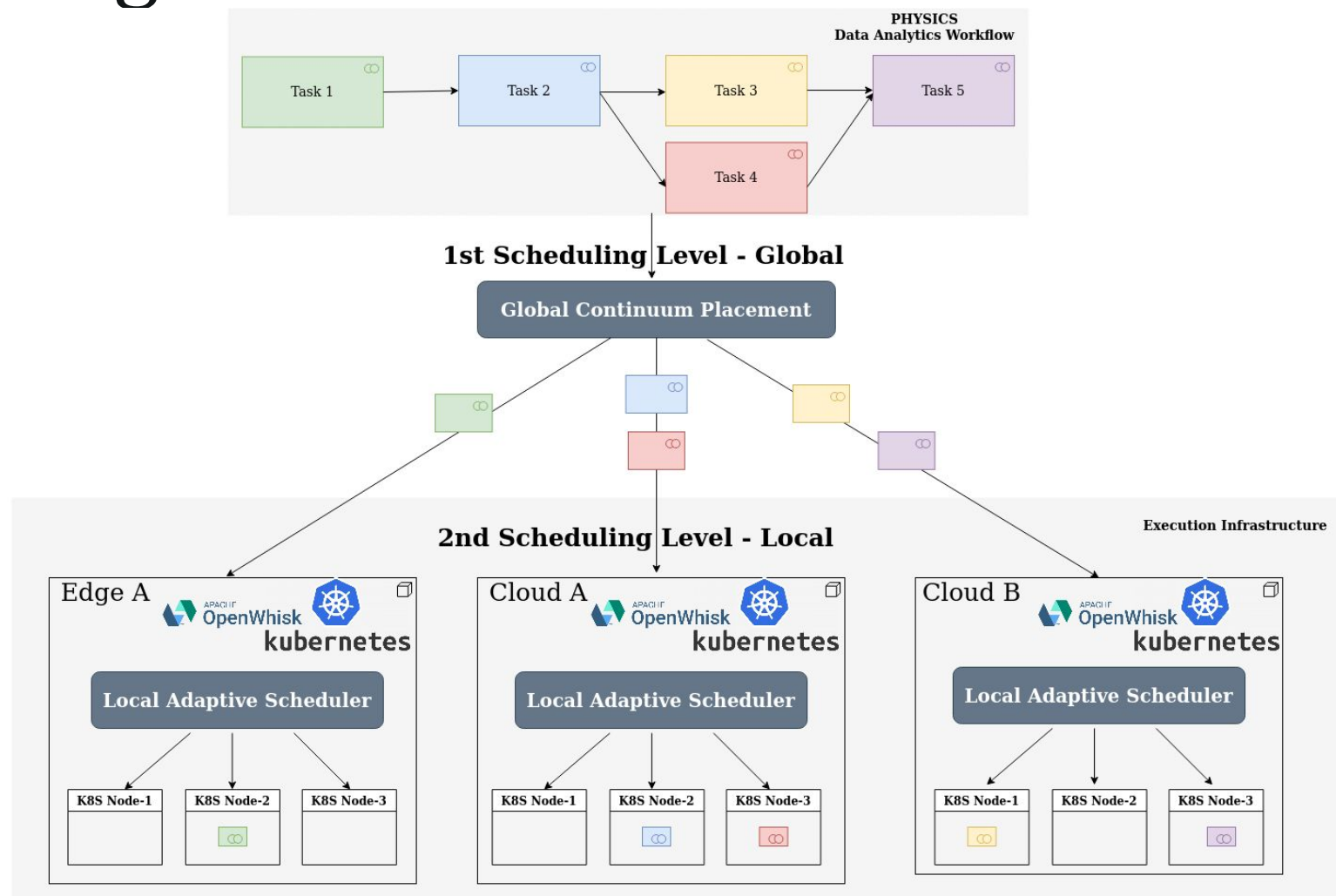
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047





Two-level scheduling for the continuum

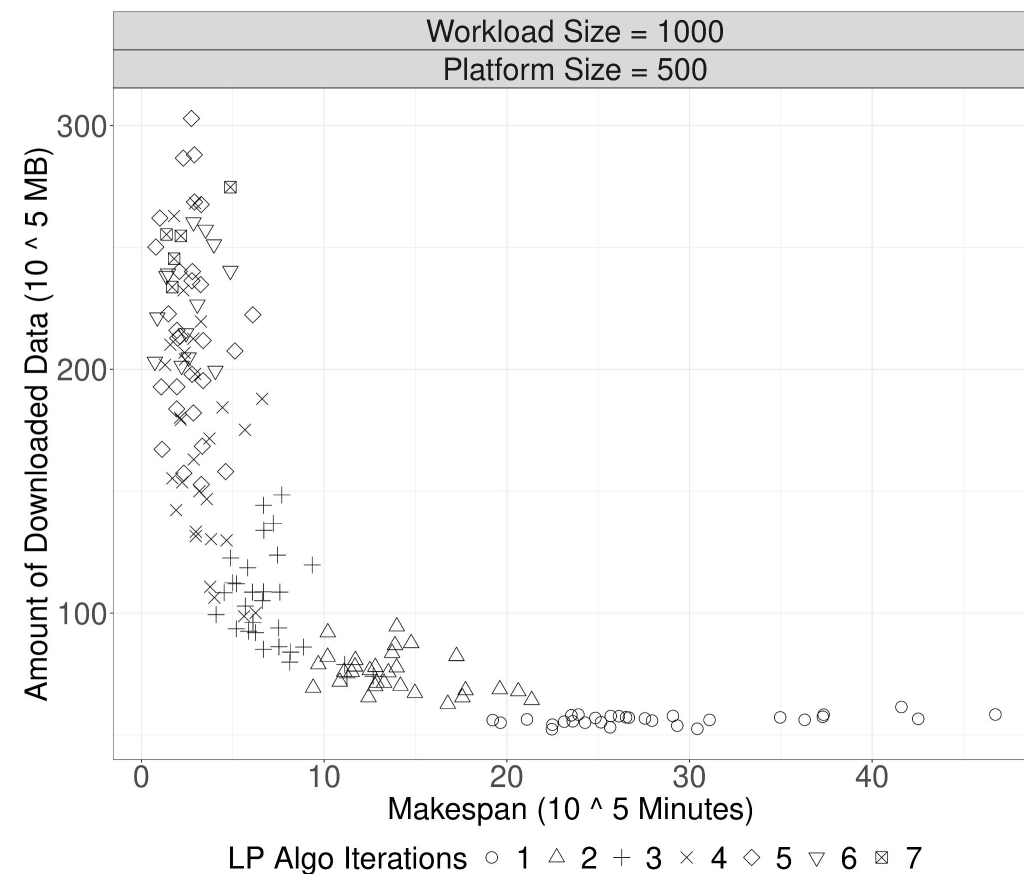
- ❖ Proposing a two-level scheduling procedure for the placement of the application components to available and suitable candidate cloud and Edge resources:
 - ❖ **1st Scheduling Level - Global** allows the selection of most adapted cluster considering aspects such as performance, energy, etc
 - ❖ **2nd Scheduling Level - Local** enables the selection of most adapted node considering FaaS related optimizations, etc





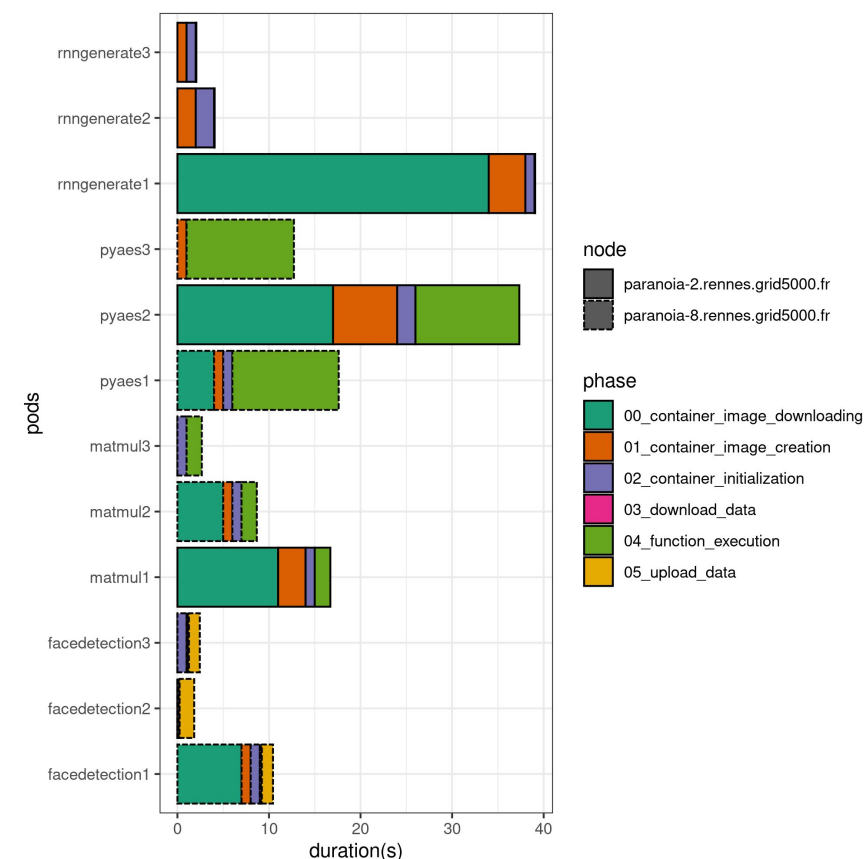
1st level - Global Continuum Placement

- ❖ Implemented our first version of Global Continuum Placement component:
 - ❖ as a multi-objective scheduler allowing multiple objectives such as: Performance, Energy, etc
 - ❖ allowing the expression of both constraints and resources needs
 - ❖ the code is published as open-source:
<https://github.com/RyaxTech/global-continuum-placement>
- ❖ Currently working on multi-objective optimizations based on **linear programming**
 - ❖ trying to minimize **makespan** and the **amount of downloaded** data by containers and function's I/O while considering platform heterogeneity
 - ❖ adaptable to various objectives
 - ❖ an article is being prepared to be submitted to CCGrid-2023



2nd level - Local Adaptive Scheduling

- ❖ Focus in intelligent scheduling algorithms for efficient resource sharing (CPU, memory, storage, network), load balancing and managing resources for FaaS execution.
- ❖ Based on the results of our studies using simulations and observations
 - ❖ Our first algorithm aims to minimize the delays due to image downloading for function execution: **minimizing Cold starts**
 - ❖ Kubernetes already provides an ImageLocality plugin
 - ❖ We have implemented a variation of ImageLocality taking into account the existence of Containers' Layers and trying to favor the execution of functions on nodes where layers of the containers to be deployed exist already. The new scheduler is named **LayersLocality**



Layers Locality Scheduling Implementation

- ❖ For this implementation we need the following:
 - ❖ To get the available layers on each node (name and size)
 - ❖ For each new pod compute a score per node considering the cumulative size of already available layers
- ❖ For this we had to make changes on various areas of the different involved software without breaking retro-compatibility:
 - ❖ Kubernetes internal interfaces to add the Layers info into Kubernetes: optional Layers field in the `core.v1.Node.NodeStatus` API
 - ❖ Container Runtime CRI-O adapted to get available layers name and size on node and send it through annotations (no API change)



Towards pushing Layers Locality Scheduling in upstream K8S

- ❖ Open-source code and installation documentation available online on github
<https://github.com/RyaxTech/k8s-container-layer-locality>
<https://github.com/RyaxTech/kubernetes>
<https://github.com/RyaxTech/cri-o>
- ❖ Discussions have been started with “sig-scheduling” Kubernetes group to push the new “Layers Locality” scheduler to the upstream Kubernetes.
- ❖ Currently working on experiments to compare the new scheduler with the typical Kubernetes scheduler and show the performance improvement for FaaS applications.





PHYSICS

Thank you for the attention!



www.physics-faas.eu



[linkedin.com/company/physicsh2020](https://www.linkedin.com/company/physicsh2020)



<https://twitter.com/H2020Physics>



info@physics-faas.eu



This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047